

**MARCUS VINÍCIUS
AMERICANO DA COSTA FILHO**

**UMA FERRAMENTA DE APOIO AO PROJETO
DE SISTEMAS DE CONTROLE PARA
PROCESSOS COM ATRASO**

**FLORIANÓPOLIS
2008**

**UNIVERSIDADE FEDERAL DE SANTA
CATARINA**

**PROGRAMA DE PÓS-GRADUAÇÃO
EM ENGENHARIA ELÉTRICA**

**UMA FERRAMENTA DE APOIO AO PROJETO
DE SISTEMAS DE CONTROLE PARA
PROCESSOS COM ATRASO**

Dissertação submetida à
Universidade Federal de Santa Catarina
como parte dos requisitos para a
obtenção do grau de Mestre em Engenharia Elétrica.

**MARCUS VINÍCIUS
AMERICANO DA COSTA FILHO**

Florianópolis, Maio de 2008.

Uma Ferramenta de Apoio ao Projeto de Sistemas de Controle para Processos com Atraso

Marcus Vinícius Americano da Costa Filho

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia Elétrica, Área de Concentração em *Controle, Automação e Informática Industrial*, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.

Julio Elias Normey Rico, Dr.
Orientador

Kátia Campos de Almeida, Ph.D.
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica

Banca Examinadora:

Julio Elias Normey Rico, Dr.
Presidente

Bismark Claure Torrico, Dr.

Aguinaldo Silveira e Silva, Ph.D.

Eugênio de Bona Castelan Neto, Dr.

Na vida nada se perde, só ganha! Para ganhar ou vencer, bastam o saber e o viver!

O Autor

*Aos meus pais, Marcus Vinícius e Maria Hermínia,
pelos quais tenho muito mais que admiração e respeito;*

*À minha irmã, Fernanda,
pelo nosso carinho mútuo;*

*Aos meus avós, Jaime e Judite Martins,
símbolos expressivos de amor, sacrifício e honradez;*

*Ao meu tio, Jaime Sérgio que,
em virtude da nossa amizade, consideramo-nos irmãos.*

AGRADECIMENTOS

Agradeço a Deus que me deu a bênção da vida e me presenteou com a liberdade de fazer escolhas; aos meus eternos amados pais que sempre me apoiaram e ensinaram a viver com dignidade, incentivando meus ideais; aos meus avós, especialmente, Jaime e Judite Martins, exemplos de força e bondade, pela presença e, sobretudo, pelo amor incondicional; à minha irmã, Fernanda, pelo companheirismo e afeto; e aos ausentes queridos que, honrosamente, cumpriram a missão outorgada pela lei divina na Terra.

Ao Prof. Dr. Julio Elias Normey Rico pela orientação, motivação e amizade.

Ao colega Dr. Bismark Claure Torrico pelas correções e valiosas sugestões.

Ao amigo Douglas Wildgrube Bertol pelo auxílio necessário.

A todos os professores do DAS, mas, principalmente, aqueles que sempre permitiram uma relação de cordialidade, proximidade e amizade com os alunos: Dr. Edson Roberto de Pieri, Dr. Eugênio de Bona Castelan Neto, Dr. Ubirajara Franco Moreno e Dr. Werner Kraus Junior.

À UFSC e à CAPES pelo suporte material e financeiro.

Aos amigos e funcionários do DAS e do PGEEL.

Enfim, a todos os meus familiares e amigos que, apesar da distância, sempre souberam me transmitir apoio, compreensão e estímulo em mais uma etapa da minha vida.

Resumo da Dissertação apresentada à UFSC como parte dos requisitos necessários para obtenção do grau de Mestre em Engenharia Elétrica.

Uma Ferramenta de Apoio ao Projeto de Sistemas de Controle para Processos com Atraso

Marcus Vinícius Americano da Costa Filho

Maio/2008

Orientador: Julio Elias Normey Rico, Dr.

Área de Concentração: Automação e Sistemas.

Palavras-chave: Ferramenta interativa, Sistemas com atraso, Identificação, Simulação, PID, Compensadores de atraso, Controle preditivo.

Número de Páginas: 86.

Este trabalho apresenta uma nova ferramenta de identificação, análise e projeto de controladores para sistemas lineares e monovariáveis com atraso de transporte, o qual integra as principais estruturas de controle utilizadas no meio industrial: PID (*Proportional-Integral-Derivative*), DTC (*Dead-time Compensator*) e MPC (*Model Predictive Control*).

A ferramenta, que permite o estudo de várias estruturas de controle, o entendimento das mesmas e a comparação de resultados obtidos, foi desenvolvida com a finalidade de dar suporte aos algoritmos de controle estudados no livro *Control of Dead-time Processes* e suprir a falta de simuladores deste tipo, especialmente orientados aos sistemas com atraso.

No decorrer do texto, o leitor terá acesso aos principais conceitos relacionados ao tema com diversos exemplos aplicados. Conceitos estes que foram fundamentais para o desenvolvimento do projeto.

Um estudo de caso industrial também faz parte do trabalho. Em parceria com a usina *Ingenio La Unión, S.A.*, localizada na Guatemala, a ferramenta foi utilizada para a modelagem, projeto e simulação dos laços de controle de seus processos, onde, com experimentos realizados, foi constatada uma grande melhora no desempenho de seus sistemas.

Ademais, está vinculado ao projeto, um manual que descreve detalhadamente as funções e configurações da ferramenta, permitindo seu uso de forma fácil e correta.

Abstract of Dissertation presented to UFSC as a partial fulfillment of the requirements for the degree of Master in Electrical Engineering.

A Tool to Support the Project of Control Systems for Processes with Delay time

Marcus Vinícius Americano da Costa Filho

May/2008

Advisor: Julio Elias Normey Rico, Dr.

Area of Concentration: Automation and Systems.

Key words: Interactive tool, Systems with delay time, Identification, Simulation, PID, Dead-time compensators, Predictive control.

Number of Pages: 86.

This work presents a new tool for identification, analysis and design of controllers for linear and monovariable systems with delay time which integrates the main control structures used in the industrial environment: PID (Proportional-Integrative-Derivative), DTC (Dead-time Compensator) and MPC (Model Predictive Control).

The tool which allows the study of many control structures, the understanding of them and the comparison of results, was developed to support the control algorithms studied in the book Control of Dead-time Processes and supply the lack of simulators of this type, especially targeted to systems with delay time.

In the text, the reader will have access to the main concepts related to the subject with several examples applied. Those concepts were fundamental to the development of the project.

An industrial case study is also part of the work. In a partnership with the plant Ingenio La Union, S.A., located in Guatemala, the tool was used to model, design and simulate the mesh of control of its processes. With the realization of experiments, it was verified a great improvement in performance of their systems.

Moreover, a handbook that describes in detail the functions and settings of the tool is bound to the project, allowing its easy and correct use.

Sumário

1	Introdução	1
1.1	Processos com atraso	1
1.2	Controle de processos com atraso	5
1.3	Ferramentas de auxílio à análise e projeto	6
1.4	Objetivos	7
1.5	Organização do trabalho	7
2	Controle de processos com atraso	9
2.1	Problemas de controle de sistemas com atraso	9
2.2	Identificação de sistemas	12
2.2.1	Modelos contínuos	13
2.2.2	Modelo discreto	15
2.2.3	Aproximação de Padé	16
2.3	Controladores PID para processos com atraso	18
2.3.1	Ajuste do PID pelo controlador equivalente do PS	19
2.4	Controladores DTC	23
2.5	Controladores MPC	31
2.5.1	O algoritmo GPC	33
2.5.2	O Compensador de atraso baseado no GPC	37

3	A Ferramenta	41
3.1	Estrutura	41
3.1.1	Aspectos de implementação	43
3.1.2	Ferramenta de Identificação	44
3.1.3	Ferramenta de Controle	48
3.2	Aplicação	50
3.2.1	Identificação do processo	50
3.2.2	Controlador PID	52
3.2.3	Controlador DTC-GPC	55
4	A Ferramenta aplicada em projetos de controle em uma usina de produção de açúcar	58
4.1	Introdução	58
4.2	Projeto do controle de nível	60
4.2.1	Modelagem do nível do cristalizador	61
4.2.2	Modelagem do conjunto bomba-variador de frequência	62
4.2.3	Modelagem do sensor de nível	62
4.2.4	Simulação	64
4.2.5	Experimentos realizados	67
4.3	Projeto do controle de temperatura	68
4.3.1	Modelagem e identificação do sistema	69
4.3.2	Simulação	71
5	Conclusões	73

A	Manual	75
A.1	Objetos de uma interface gráfica	75
A.2	Instalando e iniciando a ferramenta	76
A.3	Ferramenta de identificação	76
A.3.1	Método Gráfico	77
A.3.2	Métodos dos Mínimos Quadrados e dos Mínimos Quadrados Recursivo	78
A.4	Ferramenta de Controle	78
A.4.1	Controlador PID	79
A.4.2	Preditor de Smith	80
A.4.3	Observador de Perturbações	80
A.4.4	Preditor de Smith Filtrado	80
A.4.5	GPC	81
A.4.6	DTC-GPC	81
A.4.7	Salvando resultados	81
A.5	Comparando resultados finais	81

Lista de Figuras

1.1	Sistema hidráulico com atraso	2
1.2	Sistema de Laminação	3
1.3	Dornas conectadas em série	4
1.4	Resposta ao degrau do nível da 6 ^a dorna	5
2.1	Estrutura equivalente do PS	10
2.2	Respostas ao degrau do processo real e dos modelos de 1 ^a e 2 ^a ordens	15
2.3	Respostas ao degrau dos modelos contínuo e discreto	16
2.4	Aproximação de Padé	17
2.5	Respostas ao degrau em malha fechada para $L_1 = 0.5$, $L_2 = 8$ e $T_o = 0.4$. . .	22
2.6	Respostas ao degrau em malha fechada para $T_o = 0.4$ e $T_o = 2.8$	22
2.7	Respostas ao degrau em malha fechada do PS e do PID para $L = 8$	22
2.8	Estrutura para análise do DO-DTC	25
2.9	Estrutura para implementação do DO-DTC	27
2.10	Resposta do sistema com uma perturbação do tipo rampa em $t=80s$	28
2.11	Representação unificada de DTCs discretos	30
2.12	Estrutura Geral do MPC	32
2.13	Estratégia de controle preditivo GPC	34
2.14	Diagrama de blocos do GPC	36

3.1	Interface principal	42
3.2	Método gráfico	45
3.3	Comparação de resultados	49
3.4	Interfaces de referência e perturbação	50
3.5	Função de transferência da perturbação	50
3.6	MMQ Recursivo	51
3.7	Validação	52
3.8	Simulação PID	53
3.9	Simulação do DTC-GPC	56
3.10	Resultados	56
4.1	Processo de cristalização	59
4.2	Cristalizador e componentes	60
4.3	Modelo dinâmico do nível de cada cristalizador	62
4.4	Ensaio de identificação da dinâmica do nível	63
4.5	Diagrama de blocos do controle de nível de cada cristalizador	64
4.6	Diagrama de blocos equivalente do controle de nível	65
4.7	Simulação do controle de nível	67
4.8	Controle de nível anterior	67
4.9	Controle de nível atual	68
4.10	Rejeição de perturbações no cristalizador	69
4.11	Resposta ao degrau	70
4.12	Dados de entrada	70
4.13	Identificação da dinâmica da temperatura	71

4.14 Ajuste PS	72
4.15 Comparação	72
A.1 Elementos de uma interface	76

Siglas

2DOF	Two Degree of Freedom
CARIMA	Controlled Auto-Regressive Integrated Moving Average
DMC	Dynamic Matrix Control
DO-DTC	Disturbance Observer Dead-time Compensator
DTC	Dead-time Compensator
DTC-GPC	Dead-time Compensator Generalized Predictive Control
FOPDT	First Order Plus Dead-time
FSP	Filtered Smith Predictor
GPC	Generalized Predictive Control
IAE	Integrated Absolute Error
IMC	Internal Model Control
IPDT	Integrative Plus Dead-time
ISE	Integrated Square Error
MAC	Model Algorithm Control
MPC	Model Predictive Control
PI	Proportional-Integrative Controller
PID	Proportional-Integrative-Derivative Controller
PRBS	Pseudo-random Binary Signal
SOPDT	Second Order Plus Dead-time
PS	Preditor de Smith

Capítulo 1

Introdução

Este trabalho apresenta uma nova ferramenta para a simulação, identificação, análise e projeto de controladores para sistemas com atraso. A ferramenta integra as principais técnicas de identificação e estruturas de controle mais utilizadas no meio industrial: métodos gráficos de identificação pela resposta ao degrau, o método de identificação pelos mínimos quadrados, controladores PID (proporcional-integral-derivativo), preditor de Smith e compensadores de tempo morto (DTC - *Dead-time Compensators*) e controle preditivo (MPC - *Model Predictive Controller*). A ferramenta permite o estudo de várias estruturas, o entendimento das mesmas e a comparação de resultados com diferentes ajustes de controladores e situações de operação.

1.1 Processos com atraso

Atrasos (ou tempos mortos) são encontrados em muitos sistemas dinâmicos. Diversos processos industriais apresentam atraso na dinâmica que relaciona as variáveis manipuladas e controladas. Outros sistemas, enquadrados fora do ambiente industrial, como sistemas biológicos, econômicos, sociais, etc., também apresentam dinâmicas com atraso quando são interpretados como sistemas entrada-saída. O atraso é causado, principalmente, pelo transporte de informação, massa ou energia dentro do sistema. Porém, ele também pode ser causado pelo próprio tempo de reação do processo, do sensor ou atuador. Em outros processos, o atraso observado na dinâmica do sistema é gerado em decorrência do efeito produzido pela resposta no tempo de um determinado número de sistemas dinâmicos conectados em série [25].

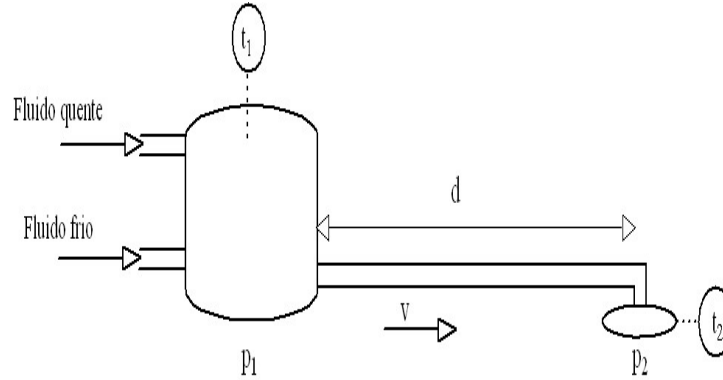


Figura 1.1: Sistema hidráulico com atraso

Muitos sistemas simples e cotidianos também apresentam atraso de transporte. Considera-se, por exemplo, uma instalação hidráulica residencial com tanques de água quente e fria localizados a uma certa distância do local de utilização da água. É comum que o usuário sinta uma certa dificuldade em ajustar uma temperatura: abre-se em excesso a torneira (ou válvula) de água quente (ou fria), procurando atingir a temperatura desejada mais rapidamente, sem perceber que o efeito da manipulação da mesma só será notado depois de um tempo. Assim, normalmente, a água esquenta ou esfria demais e precisa-se de algumas iterações até atingir o valor desejado. Isto se deve ao atraso de transporte que impede, inicialmente, que os ajustes feitos nas válvulas de água sejam percebidos, instantaneamente, pelo usuário.

Um sistema similar ao domiciliar é ilustrado na figura 1.1. Trata-se de um tanque de mistura de dois fluidos, um quente e um frio, e uma tubulação de comprimento d que leva o fluido misturado até o ponto de utilização (p_2). Se a temperatura das vazões combinadas é medida no ponto p_1 , pode-se considerar que a temperatura da mistura se modifica imediatamente após a mudança da vazão de um dos fluidos. Entretanto, a temperatura do fluido no ponto p_2 não muda instantaneamente, pois o fluido com temperatura modificada necessita de um certo tempo para se mover fisicamente até o ponto de medição. O tempo de atraso pode ser calculado por:

$$L = \frac{d}{v} \text{ (segundos)} \quad (1.1)$$

onde d é a distância entre os pontos de mistura e medição, em metros; e v é a velocidade do fluxo da mistura, em m/seg.

Assumindo-se que $t_1(t)$ é a temperatura no ponto de mistura e que a tubulação é perfeita (não há perdas neste trajeto), tem-se:

$$t_2(t) = t_1(t - L) \quad (1.2)$$

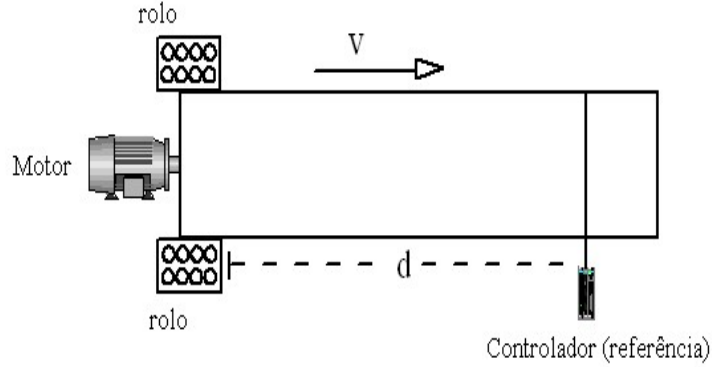


Figura 1.2: Sistema de Laminação

onde $t_2(t)$ é a temperatura no ponto de medição. Considerando a tubulação como um sistema entrada-saída, observa-se que a saída $t_2(t)$ está deslocada L segundos em relação à entrada $t_1(t)$.

Aplicando a transformada de Laplace na equação (1.2), obtém-se:

$$T_2(s) = e^{-Ls}T_1(s) \quad (1.3)$$

A equação (1.3) representa o modelo entrada-saída no domínio s , sendo o atraso de transporte representado pelo termo e^{-Ls} .

Um sistema físico com um princípio semelhante é o de laminação, onde um servo-motor é usado para ajustar a separação dos rolos compressores para se obter a espessura desejada da lâmina, como é mostrado na figura 1.2. Se $w(t)$ é a espessura da lâmina na saída dos rolos e $w_m(t)$ é a espessura medida a uma certa distância d dos mesmos, existirá um atraso de transporte dado por $L = d/v$ entre as duas medidas, onde v é a velocidade de movimentação da lâmina:

$$\begin{aligned} w_m(t) &= w(t - L) \\ W_m(s) &= e^{-Ls}W(s) \end{aligned}$$

Como dito anteriormente, sistemas dinâmicos conectados em série podem causar um efeito similar ao do atraso na relação dinâmica entre a entrada e saída do processo. Considere-se, por exemplo, um conjunto de dornas cilíndricas conectadas em série de acordo com a figura 1.3, onde o nível de cada uma esteja próximo ao ponto de operação. O comportamento dinâmico do nível de cada dorna pode ser modelado por um sistema linear:

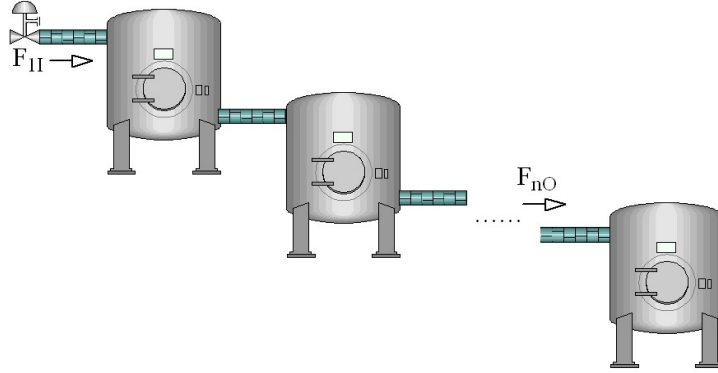


Figura 1.3: Dornas conectadas em série

$$\begin{aligned} A \frac{dH_i}{dt} &= F_{iI} - F_{iO} \\ F_{iO} &= KH_i \end{aligned} \quad (1.4)$$

onde H_i , F_{iI} e F_{iO} são, respectivamente, o nível, a vazão de entrada e de saída da dorna i , A é a área da base da dorna e K é uma constante que depende das características da dorna. Aplicando a transformada de Laplace na equação (1.4) e considerando suas condições iniciais nulas, tem-se:

$$\begin{aligned} AsH_i(s) &= F_{iI}(s) - KH_i(s) \\ H_i(s) &= \frac{1/K}{Ts+1} F_{iI}(s), \quad T = A/K \end{aligned}$$

Para a 1ª dorna:

$$H_1(s) = \frac{1/K}{Ts+1} F_{1I}(s)$$

e para a 2ª dorna:

$$H_2(s) = \frac{1/K}{Ts+1} F_{2I}(s)$$

Nota-se que o fluxo de entrada da 2ª dorna é o fluxo de saída da 1ª dorna e que substituindo, obtém-se:

$$H_2(s) = \frac{1/K}{(Ts+1)^2} F_{1I}(s)$$

Se este procedimento for aplicado sucessivamente, a função de transferência, $P(s)$, que relaciona a entrada, $F_{1I}(s)$, e a saída, $H_n(s)$, da n -ésima dorna é:

$$P(s) = \frac{1/K}{(Ts+1)^n}$$

A resposta ao degrau deste sistema apresenta uma característica que se assemelha à um atraso

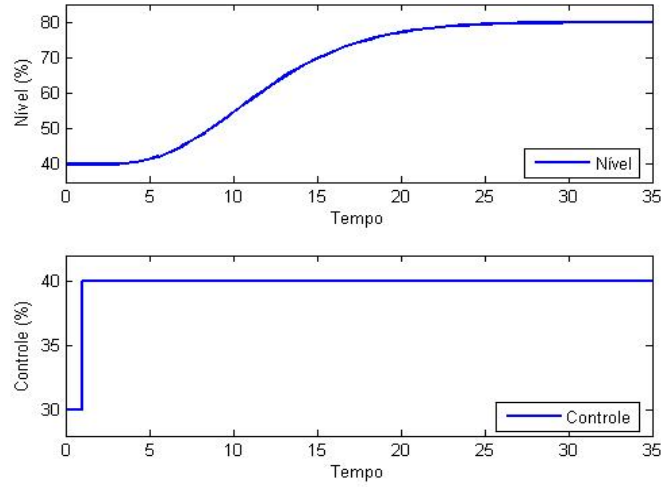


Figura 1.4: Resposta ao degrau do nível da 6ª dorna

de transporte, como se ilustra na continuação. Seja um sistema de 6 dornas conectadas em série onde considera-se $K = 1$ e $T = 2$ segundos:

$$P(s) = \frac{1}{(2s + 1)^6}$$

Aplicando um degrau de 10% na entrada do sistema em malha aberta, observa-se uma variação de 40% no nível da dorna e um atraso aparente de aproximadamente 3 segundos (ver figura 1.4), apesar de que o modelo de $P(s)$ não possui atraso. Este atraso aparente é causado pela “lentidão” da resposta do sistema de alta ordem.

Este comportamento é bastante observado em processos industriais, motivo pelo qual o modelo do atraso é muito usado para representar parte da dinâmica de muitas plantas. O modelo de primeira ordem com atraso é o mais utilizado na indústria de processos.

1.2 Controle de processos com atraso

Processos com atrasos são difíceis de controlar por controladores convencionais, principalmente porque ocorre um atraso tanto na percepção dos efeitos das perturbações quanto no efeito da ação de controle na variável controlada. Em outras palavras, a ação de controle atual, além de ser baseada num evento que ocorreu no passado, leva um determinado tempo para ser notada na saída do sistema.

No domínio da frequência, observa-se que o atraso diminui muito rapidamente a fase do sistema, principalmente em altas frequências, tornando difícil o ajuste do controlador. Quando o atraso é significativo, um controle convencional deverá ser ajustado para se obter uma resposta lenta o suficiente, de tal forma que permita que a ação do avanço de fase do mesmo compense a queda de fase introduzida pelo atraso [25].

Dentro do contexto do controle de processos industriais, os algoritmos dedicados ao controle de sistemas com atraso são bastante importantes. Isto é devido ao fato de que o atraso de transporte está presente na dinâmica da maioria dos processos industriais. Muitos trabalhos têm sido apresentados abordando este assunto nos últimos anos, mostrando sua importância acadêmica [28].

Controladores PID, presentes em todo tipo de indústria, podem ser usados para este tipo de processo. No entanto, se o atraso for muito grande, o ajuste do controlador já não é conveniente e um desempenho pior pode ser apresentado pelo sistema. Desta forma, quando se deseja respostas mais rápidas, devem ser usados controladores que incluam a compensação do atraso.

Dois grupos de controladores que incluem compensação de atraso são os mais importantes em nível industrial: os DTCs e os MPCs. A sua importância pode ser verificada através de pesquisas que mostram que eles são os controladores mais utilizados na indústria depois dos PIDs. Sendo assim, o estudo de técnicas de controle de sistemas com atraso, baseadas em PID, DTC e MPC, é de grande importância tanto em nível acadêmico como industrial [18],[5].

1.3 Ferramentas de auxílio à análise e projeto

A simulação é uma técnica amplamente usada para avaliar o desempenho de sistemas, aumentando a produtividade do engenheiro assim como incentivando sua análise e estudo na aplicação de novas técnicas de controle. No âmbito mundial, o desenvolvimento de ferramentas de análise e projeto à disposição do engenheiro de controle vem experimentando, nestas últimas cinco décadas, uma evolução admirável na busca do estado da arte nesta área [35],[4],[11], [10]. À medida que novas metodologias e aplicações na teoria de controle são desenvolvidas, cresce a demanda por ferramentas computacionais, tornando a CACE (*Computer Aided Control Engineering*) uma das principais áreas de pesquisa em controle automático.

Ou seja, as ferramentas da CACE são muito importantes para que o usuário (estudante ou engenheiro) possa comparar controladores com diferentes graus de complexidade na tomada de decisão e escolha da alternativa mais adequada para um determinado processo. Para a indústria, isto significa o melhor balanceamento possível do “custo X benefício”.

Como comentado, em controle de processos industriais, os algoritmos dedicados ao controle de sistemas com atraso mais utilizados são os PIDs, DTCs e MPCs. Na atualidade, estas opções estão disponíveis para o engenheiro em plataformas de controle comerciais. A questão é saber qual controlador apresenta um melhor resultado para um determinado modelo de processo e para uma determinada situação de operação. Por isso o desenvolvimento de uma ferramenta de apoio para o estudo deste tipo de sistema se torna necessário.

1.4 Objetivos

Este trabalho tem como objetivo o desenvolvimento de uma ferramenta de apoio para o estudo, simulação e projeto de controladores para sistemas que contêm atraso de transporte. A ferramenta desenvolvida neste trabalho está direcionada, na sua primeira versão, para processos monovariáveis. O foco da ferramenta é servir como auxílio no estudo do livro *Control of Dead-time Processes* [25] onde diversos algoritmos de identificação e controle são apresentados. A ferramenta está disponível no site: <http://www.das.ufsc.br/~julio/deadtimebook.html>.

1.5 Organização do trabalho

O texto do trabalho foi elaborado respeitando uma sequência lógica relacionada ao assunto e todas as explicações necessárias foram fornecidas no desenvolvimento do projeto. Sucintamente, os principais problemas de controle de sistemas com atraso de transporte e suas soluções, utilizando o PID, DTC e MPC, são expostos no capítulo 2. Os principais modelos utilizados na indústria (contínuo e discreto) obtidos a partir de técnicas de identificação são apresentados neste capítulo, assim como são analisadas as propriedades, métodos de ajuste e implementações dos controladores.

No capítulo 3, a ferramenta é apresentada, focalizando o texto na descrição de sua estrutura, navegação, aplicações e nos seus principais recursos. São ilustrados diversas interfaces gráficas que compõem a ferramenta e algumas particularidades de implementação, tanto

no pacote de identificação quanto no de controle, os quais facilitam bastante a interação com o usuário.

No capítulo 4, a ferramenta é aplicada em um caso industrial. É feita uma breve descrição de como é o processo de fabricação de açúcar e, em parceria com a usina *Ingenio La Unión, S.A.*, localizada na Guatemala, a ferramenta é utilizada para a modelagem, projeto e simulação dos laços de controle de seus processos.

Por fim, um manual que descreve detalhadamente as funções e configurações da ferramenta, permitindo seu uso de forma fácil e correta, encontra-se no apêndice deste documento.

Capítulo 2

Controle de processos com atraso

Este capítulo tem como objetivo apresentar de forma resumida os principais problemas de controle de sistemas com atraso e suas soluções propostas na literatura. Aspectos teóricos da identificação e do controle destes sistemas utilizando PID, DTC e MPC também serão apresentados.

2.1 Problemas de controle de sistemas com atraso

Como citado anteriormente, é difícil obter um bom desempenho quando são utilizados controladores convencionais para controlar processos com atraso. Apesar deste fato, a maioria dos métodos de ajuste de controladores PID usados na indústria é baseada em modelos que incluem um atraso na dinâmica do processo. Muitos trabalhos analisam o ajuste de PIDs para estes sistemas, sempre obtendo um ajuste que gere uma resposta lenta o suficiente para permitir um certo grau de robustez do sistema em malha fechada [25]. Quando deseja-se respostas mais rápidas, um compensador de atraso é introduzido na malha de controle.

O Preditor de Smith [38], mostrado na figura 2.1, é sem dúvida o método mais usado na indústria para controlar sistemas com atraso e pode ser considerado como o primeiro método de controle com compensação de atraso apresentado na literatura. Nesta representação, $P_n(s) = G_n(s)e^{-L_n s}$ é o modelo do processo, $G_n(s)$ é o modelo rápido (sem atraso) e $C(s)$ é o controle primário.

A principal vantagem do PS é que o atraso de transporte é eliminado da equação

característica de malha fechada do sistema,

$$1 + C(s)G_n(s) = 0,$$

e a função de transferencia entrada-saída nominal resulta:

$$H_r(s) = \frac{Y(s)}{R(s)} = \frac{C(s)G_n(s)e^{-L_n s}}{1 + C(s)G_n(s)} \quad (2.1)$$

Observa-se, que no caso nominal, o atraso não afeta o cálculo de $C(s)$ para obter uma $H_r(s)$ desejada.

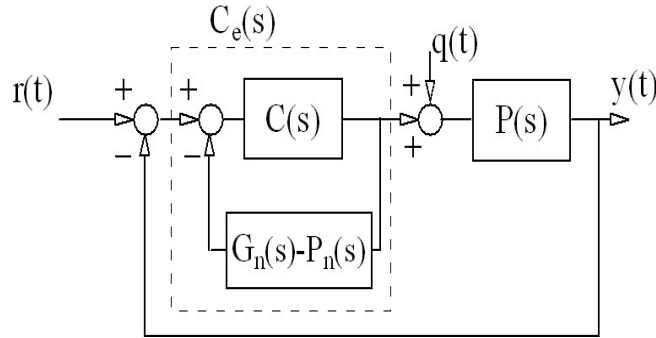


Figura 2.1: Estrutura equivalente do PS

Uma leitura superficial das propriedades do PS pode levar a pensar que o ajuste do controle primário, $C(s)$, poderia ser realizado considerando apenas o modelo da planta sem atraso e que a velocidade da resposta em MF poderia ser arbitrada. Porém, como todo controlador, os erros de modelagem devem ser considerados para obter um ajuste robusto do PS.

Outras limitações estão relacionadas com a estrutura do PS e a rejeição de perturbações: o sistema de controle não pode ser utilizado com processos que tenham pólos com parte real positiva e, se o processo for integrador, o PS não rejeita perturbações de carga constante na entrada do processo [24, 31].

Também, mesmo no caso de plantas estáveis, não é possível definir arbitrariamente o comportamento do sistema frente às perturbações de carga, ajustando somente o controle primário. Estas propriedades podem ser estudadas a partir da função de transferência perturbação-saída:

$$H_q(s) = \frac{Y(s)}{Q(s)} = P_n(s) \left[1 - \frac{C(s)P_n(s)}{1 + C(s)G_n(s)} \right] \quad (2.2)$$

onde observa-se que os pólos de $P_n(s)$ aparecem em $H_q(s)$, independentemente do ajuste de $C(s)$. Isto explica porque os pólos lentos ou instáveis da planta não podem ser eliminados de $H_q(s)$.

Para considerar os erros de modelagem, usar-se-á o modelo: $P(s) = P_n(s)(1 + \delta P(s))$, onde $\delta P(s)$ é o erro multiplicativo. Assim, o módulo do máximo erro de modelagem multiplicativo admissível para estabilidade robusta é [20]:

$$dP(\omega) = \frac{|1 + C(j\omega)G_n(j\omega)|}{|C(j\omega)G_n(j\omega)|} \quad \forall \omega > 0 \quad (2.3)$$

onde $dP(\omega)$ pode ser usado como um índice de robustez do sistema. Nota-se que $dP(\omega) = \frac{1}{|H_r(j\omega)|}$. Assim, se $C(s)$ é usado para conseguir respostas muito rápidas, o sistema terá um índice de robustez pequeno.

Durante os últimos anos, a comunidade de controle tem avançado bastante no sentido de solucionar as limitações do PS, propondo diversas modificações que podem ser agrupadas no conjunto dos DTCs [47],[23],[29],[27],[46],[22],[14],[31]. Algumas delas serão discutidas neste capítulo.

Os controladores preditivos também são algoritmos muito utilizados na indústria para controlar processos com atrasos. Como é mostrado em [25], estes algoritmos também podem ser analisados como compensadores de tempo morto. Logo, suas propriedades podem ser analisadas conjuntamente com as dos DTCs. Pode-se considerar que os MPCs são uma evolução dos DTCs onde novas características são introduzidas, como otimização e tratamento de restrições.

Todos os algoritmos anteriores precisam de um modelo do processo para o cálculo da lei de controle. Teoricamente, o modelo do processo poderia ser determinado, utilizando um conjunto de equações dinâmicas que representam as características físicas da planta. Porém, na prática industrial, é muito comum utilizar modelos matemáticos obtidos por técnicas de identificação baseadas em dados de entrada-saída do processo.

2.2 Identificação de sistemas

O comportamento físico de sistemas dinâmicos pode ser descrito por um conjunto de equações diferenciais. Quando isto não é possível, recorre-se à modelagem do sistema a partir de testes e experimentos. Este procedimento é conhecido como identificação de sistemas e pode ser feito, basicamente, em três etapas [21]:

- **Coleta de dados.** Nesta etapa, deve ser definido qual o tipo de sinal é o mais apropriado para aplicar na entrada para, a partir de então, obter-se a resposta do processo. Com as medidas de entrada e saída armazenadas, pode-se, posteriormente, implementar e avaliar os algoritmos de estimação não-recursivos.
- **Definição do conjunto de modelos.** Nesta etapa, um conjunto de possíveis modelos (ou estrutura do modelo) é definido. É o passo mais importante. Existem, essencialmente, dois casos:
 - 1) Quando se tem algum conhecimento a respeito da estrutura do modelo baseada numa informação física e o objetivo é encontrar os parâmetros do mesmo, usando interpretações físicas. É denominada identificação caixa-cinza.
 - 2) Quando não se tem conhecimento a respeito do processo e os parâmetros são utilizados para ajustar o modelo compatível com os dados adquiridos. É denominada identificação caixa-preta.
- **Determinação do melhor modelo do conjunto, baseado nos dados adquiridos.** Nesta etapa, é feita uma avaliação da qualidade do modelo. Consiste em checar o comportamento dos modelos quando eles tentam reproduzir os dados reais. É, usualmente, feita pela minimização de um critério que mostra o quão bem o modelo gera os dados avaliados.

Depois de executar as três etapas vistas anteriormente, deve-se escolher um modelo do conjunto que descreve da melhor forma o comportamento do processo de acordo com o critério analisado. Feito isso, o modelo deve passar pela etapa de validação que consiste numa comparação dos dados do modelo com os dados reais de maneira mais estendida. Se o modelo não passar no teste de validação, as etapas do procedimento devem ser revisadas. Isto acontece quando o conjunto de modelos não é apropriado ou o critério não é bem escolhido ou também porque os dados obtidos não são ricos o suficiente.

Diferentes tipos de modelos podem representar processos com atraso. A maioria das identificações são feitas em malha aberta, com poucas exceções [15]. Como o atraso está presente em muitos processos industriais, modelos contínuos de primeira e segunda ordens com atraso (FOPDT - *First-order Plus Dead-time*, SOPDT - *Second-order Plus Dead-time*) são usados em larga escala para o ajuste de controladores industriais. Usualmente, um modelo é estimado da resposta ao degrau de um processo em malha aberta e usado para o projeto de controladores PI e PID [3].

Neste trabalho, somente modelos lineares com atraso, representados por funções de transferência em tempo contínuo e em tempo discreto, são considerados.

2.2.1 Modelos contínuos

Modelos estáveis básicos

Dois tipos de modelo serão usados para processos estáveis: modelos FOPDT e SOPDT.

O modelo de primeira ordem é representado pela seguinte função de transferência:

$$P(s) = \frac{K_p}{Ts + 1} e^{-Ls} \quad (2.4)$$

onde a $T > 0$ é a constante de tempo, $L > 0$ é o tempo de atraso e K_p é o ganho estático do sistema.

Neste tipo de modelo, a relação L/T é muito importante. Geralmente, na indústria, define-se o atraso normalizado como:

$$\tau = \frac{L}{L + T}, \quad 0 \leq \tau \leq 1$$

e considera-se um processo com atraso dominante quando $\tau > 2/3$ ($L > 2T$). Esta medida está relacionada com a dificuldade de controle do processo. Quanto maior τ , mais difícil é o ajuste de controle. τ também é denominada *razão de controlabilidade*.

Quando se deseja representar um processo com resposta transitória mais suave ou oscilatória, um modelo de segunda ordem é utilizado:

$$P(s) = \frac{K_p e^{-Ls}}{(T_1 s + 1)(T_2 s + 1)} \quad (2.5a)$$

$$P(s) = \frac{K_p e^{-Ls}}{\frac{1}{\omega_n^2} s^2 + \frac{2\xi}{\omega_n} s + 1} \quad (2.5b)$$

onde $T_1 > 0$ e $T_2 > 0$ são as constantes de tempo para uma resposta não oscilatória; $L > 0$ é o valor do atraso; $\omega_n > 0$ é a frequência natural; $\xi \in [0, 1]$ é o coeficiente de amortecimento e K_p é o ganho estático.

Modelos com ação integrativa

Quando o processo apresenta um comportamento integrativo, dois modelos simples podem ser utilizados: Modelo integrativo de primeira ordem (IPDT - *Integrative Plus Dead-time*) e modelo integrativo de segunda ordem (SOPIDT - *Second-order Plus Integrative Dead-time*).

O modelo integrativo de primeira ordem é dado por:

$$P(s) = \frac{K_v}{s} e^{-Ls} \quad (2.6)$$

onde $L > 0$ é o valor do atraso e K_v é o ganho de velocidade.

Como no modelo estável, um modelo integrativo de segunda ordem é usado para representar processos integrativos com uma resposta transitória mais suave:

$$P(s) = \frac{K_v}{s(Ts + 1)} e^{-Ls} \quad (2.7)$$

onde $T > 0$ é o parâmetro equivalente à constante de tempo da parte não integrativa do processo.

Modelos instáveis

Alguns sistemas dinâmicos apresentam um comportamento instável em malha aberta que pode ser representado de maneira simples por:

$$P(s) = \frac{K_p}{-Ts + 1} e^{-Ls} \quad (2.8)$$

onde K_p e $T > 0$ são usados para representar o ganho e o incremento exponencial da saída do processo, respectivamente.

Normalmente, em todos estes modelos de baixa ordem, o atraso é utilizado para representar as dinâmicas de alta ordem do processo. O seguinte exemplo ilustra este caso.

Exemplo 2.1: na figura 2.2 são mostradas as respostas ao degrau do processo das seis dornas conectadas em série, visto no capítulo 1, $P(s) = \frac{1}{(2s+1)^6}$; e dos seus modelos de 1ª ordem, $P_{n1}(s) = \frac{1}{6.3s+1}e^{-6.7s}$, e de 2ª ordem, $P_{n2}(s) = \frac{1}{25s^2+9s+1}e^{-3.5s}$.

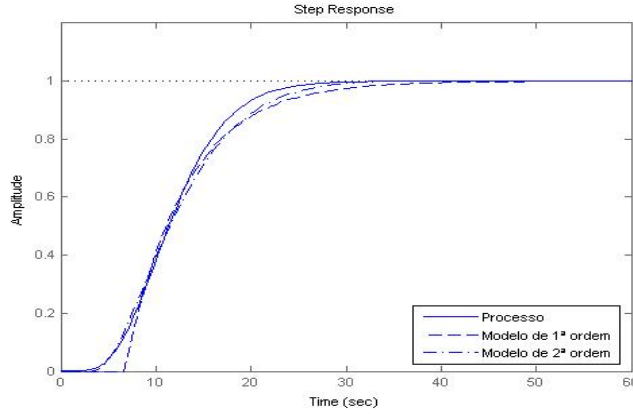


Figura 2.2: Respostas ao degrau do processo real e dos modelos de 1ª e 2ª ordens

Nota-se que o atraso pode representar dinâmicas de ordens maiores. Neste caso, o sistema é mais bem representado pelo modelo de 2ª ordem, dado que há uma melhor aproximação do atraso aparente do processo pelo atraso do modelo.

2.2.2 Modelo discreto

O modelo discreto genérico de um processo linear é dado pela seguinte forma:

$$A(z^{-1})y(t) = z^{-d}B(z^{-1})u(t-1) + e(t) \quad (2.9)$$

onde $u(t)$ e $y(t)$ são a entrada e saída, respectivamente; d é o atraso em número de amostras e $e(t)$ é o ruído. $A(z^{-1})$ e $B(z^{-1})$ são polinômios no operador argumento para trás¹, z^{-1} :

$$\begin{aligned} A(z^{-1}) &= 1 + a_1z^{-1} + a_2z^{-2} + \dots + a_naz^{-na} \\ B(z^{-1}) &= b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_nbz^{-nb} \end{aligned}$$

¹Neste trabalho, z é usado para representar tanto o operador deslocamento no tempo quanto a variável complexa no domínio da transformada z .

E o modelo do processo é relacionado com o modelo da função de transferência por:

$$P(z) = G(z)z^{-d} = \frac{B(z^{-1})z^{-1}}{A(z^{-1})}z^{-d}$$

Observa-se que, neste caso, o atraso é representado por um polinômio em z^{-1} .

Quando o período de amostragem é bem escolhido, o modelo discreto do processo obtido, considerando a dinâmica do segurador de ordem zero, pode representar adequadamente a dinâmica desejada.

Exemplo 2.2: Discretizando o modelo contínuo de 1ª ordem do exemplo 2.1 usando um segurador de ordem zero, com um período de amostragem $T_s = 0.5$, obtém-se: $P_{nd}(z) = \frac{0.0465z+0.0298}{z-0.9237}z^{-14}$. As respostas ao degrau dos modelos contínuo e discreto são mostradas na figura 2.3.

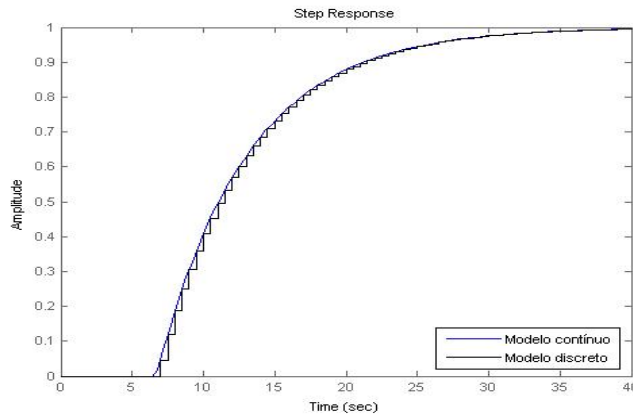


Figura 2.3: Respostas ao degrau dos modelos contínuo e discreto

2.2.3 Aproximação de Padé

Como foi apresentado nas seções anteriores, o atraso é representado por uma função não polinomial. Isto causa dificuldades operacionais em algumas situações e opta-se pela aproximação do atraso por uma função racional. A aproximação de Padé é o meio mais usado para aproximar a função e^{-Ls} por uma função racional. Consiste em igualar os primeiros termos de e^{-Ls} a uma função racional com p zeros e q pólos. Desta forma, obtém-se o que se chama de aproximação de Padé (p,q) . Geralmente, utiliza-se $p = q$. Neste caso, denomina-se aproximação de Padé de p -ésima ordem. As mais utilizadas são as de 1ª e 2ª ordens, dadas

por:

$$P_{11}(s) = \frac{1 - \frac{L}{2}s}{1 + \frac{L}{2}s}, \quad P_{22}(s) = \frac{1 - \frac{L}{2}s + \frac{L^2}{12}s^2}{1 + \frac{L}{2}s + \frac{L^2}{12}s^2}.$$

Na figura 2.4 é mostrado um exemplo, comparando as respostas ao degrau das aproximações e do sistema exato de 1ª ordem, $P(s) = \frac{1}{s+1}e^{-3s}$. O código do Matlab deste exemplo é:

Código do MATLAB®

```
% processo
num=1;den=[1 1];atraso=3;

% sem atraso
p=tf(num,den);
% com atraso
P=tf(num,den,'inputdelay',atraso);

% aproximação de Padé de 1a ordem
p11=tf([-atraso 2],[atraso 2]);
P1=p*p11;

% aproximação de Padé de 2a ordem
a2=conv(atraso,atraso);
p22=tf([a2/12 -atraso/2 1],[a2/12 atraso/2 1]);
P2=p*p22;

step(P,P1,'--',P2,'-.')
```

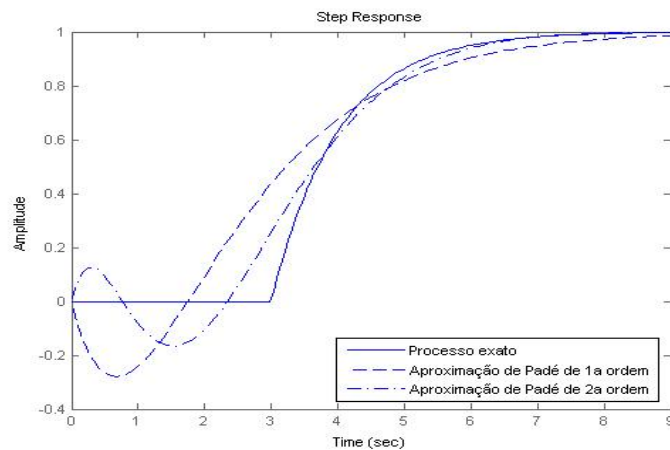


Figura 2.4: Aproximação de Padé

Nota-se que a “resposta inversa” para $t < 3$ é gerada em virtude dos zeros de fase não-mínima presentes nas aproximações. A resposta se torna melhor para $t \geq 3$, mas, a depender do atraso, o uso desta aproximação no lugar do modelo do atraso inerente ao processo pode comprometer o ajuste do controle e, conseqüentemente, o desempenho do sistema.

Na ferramenta desenvolvida neste trabalho, os modelos aqui apresentados podem ser obtidos, utilizando dados reais ou simulados. Isto será apresentado no capítulo 3.

Como comentado, três tipos de controladores são implementados na ferramenta: PID, DTC e MPC. Nas próximas seções, analisar-se-á alguns aspectos teóricos das propriedades e ajustes dos mesmos.

2.3 Controladores PID para processos com atraso

Em muitos processos industriais, a utilização do controlador PID tem sido uma estratégia de controle que tem permitido obter resultados satisfatórios durante muitos anos. Embora tenha havido um grande desenvolvimento em novas estratégias de controle, muitas indústrias ainda adotam o controlador proporcional integrativo e derivativo. Sua popularidade nos setores acadêmico e industrial é devido ao fato da simplicidade de sua estrutura e desempenho robusto em diversas plantas industriais.

A ação de controle, $u(t)$, de um controlador PID acadêmico ideal é calculada por:

$$u(t) = K_c[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt}] \quad (2.10)$$

E sua função de transferência é:

$$C(s) = \frac{U(s)}{E(s)} = K_c(1 + \frac{1}{T_i s} + T_d s) \quad (2.11)$$

onde o ajuste dos parâmetros K_c , T_i e T_d é chamado de sintonia do controlador, para o qual diversos métodos são encontrados da literatura [3], [5], [17], [36]. O mais conhecido é o método de Ziegler-Nichols [13].

Na prática, um filtro passa baixa é colocado no termo derivativo para torná-lo realizável:

$$C(s) = \frac{U(s)}{E(s)} = K_c(1 + \frac{1}{T_i s} + \frac{T_d s}{\alpha T_d s + 1}) \quad (2.12)$$

Embora a resposta em baixas frequências continue a mesma, a inclusão do filtro reduz a amplificação de ruído, onde $\alpha \in [0, 1]$ e $T_f = \alpha T_d$ representa a constante de tempo do filtro.

Além da forma acadêmica, existem outras duas formas de implementar o PID. A estrutura série é dada pela seguinte função de transferência:

$$C(s) = K_c \frac{T_I s + 1}{T_I s} \frac{T_D s + 1}{\alpha T_D s + 1} \quad (2.13)$$

e a paralela por:

$$C(s) = K_P + \frac{K_I}{s} + \frac{K_D s}{\alpha K_D s + 1} \quad (2.14)$$

A forma série é menos flexível do que a forma paralela pelo fato de haver uma interação maior entre o ajuste dos parâmetros e não permitir zeros complexos. Nota-se que, em alguns casos, é possível converter a forma em série para a forma paralela e vice-versa; e que o efeito dos parâmetros pode ser diferente de um caso para o outro. Por isto é muito importante se conhecer a implementação do controlador antes do ajuste dos parâmetros.

A depender da dinâmica do processo a ser controlado e de suas especificações exigidas, o PID pode ser uma boa escolha. Porém, existem algumas situações nas quais o PID não oferece uma resposta satisfatória, sendo necessário utilizar técnicas de controle mais sofisticadas.

Existem muitas técnicas de ajuste de PID. Neste trabalho, descreve-se uma delas que utiliza o conceito de predição.

2.3.1 Ajuste do PID pelo controlador equivalente do PS

Como visto no capítulo 1, o PS pode ser a solução ideal para o controle de plantas estáveis com atraso. Se, na prática, apenas dispõe-se do PID, uma boa opção de ajuste de parâmetros do controlador PID para modelos de processo de primeira ordem com atraso, é equiparar o PID com o PS, usando uma aproximação polinomial de Padé de 1ª ordem para o atraso.

Seja a estrutura equivalente de Smith da figura 2.1. O controlador equivalente é dado por:

$$C_e(s) = \frac{C(s)}{1 + C(s)[G_n(s) - P_n(s)]} \quad (2.15)$$

Se o modelo do processo é:

$$P_n(s) = \frac{K_p e^{-Ls}}{1+Ts}, \quad G_n(s) = \frac{K_p}{1+Ts}$$

e o controlador primário é um PI:

$$C(s) = \frac{K_1(1+T_1s)}{T_1s},$$

considerando uma modelagem perfeita do processo e ajustando o PI para cancelar o pólo de malha aberta, ou seja, $T_1 = T$, a equação característica é dada por:

$$\begin{cases} 1 + C(s)G_n(s) = 1 + \frac{K_1 K_p}{Ts} = 1 + \frac{1}{T_o s}, \\ T_o = \frac{T}{K_1 K_p} \end{cases}$$

e a função de transferência de malha fechada é:

$$\frac{Y(s)}{R(s)} = \frac{C(s)G_n(s)e^{-Ls}}{1 + C(s)G_n(s)} = \frac{e^{-Ls}}{1 + T_o s}$$

O controlador equivalente é dado por:

$$C_e(s) = \frac{\frac{K_1(1+Ts)}{Ts}}{1 + \frac{K_1 K_p}{Ts}(1 - e^{-Ls})} = \frac{K_1(1+Ts)}{Ts + K_1 K_p(1 - e^{-Ls})} \quad (2.16)$$

Nota-se que o controlador tem uma ação integral e pode ser aproximado por um PID, se o atraso for substituído por um polinômio.

Utilizando-se a aproximação de Padé de 1ª ordem, $P_{11} = \frac{1-sL/2}{1+sL/2}$, no atraso da equação (2.16), tem-se:

$$C_e(s) = \frac{K_1(1+Ts)}{Ts + K_1 K_p(1 - \frac{1-sL/2}{1+sL/2})} = \frac{K_1(1+Ts)}{Ts + K_1 K_p \frac{sL}{1+sL/2}} \quad (2.17)$$

$$C_e(s) = \frac{K_1(1+Ts)(2+Ls)}{Ts(2+Ls) + 2K_1 K_p Ls} = \frac{K_1(1+Ts)(2+Ls)}{Ts[(2+Ls) + (2/T_o)L]} \quad (2.18)$$

$C_e(s)$ pode ser representado de tal forma que se torne um PID com um filtro na parte derivativa:

$$C_e(s) = \frac{K_c(1+T_i s)(1+T_d s)}{T_i s(1+\alpha T_d s)} \quad (2.19)$$

onde $T_i = T, T_d = 0.5L$ e:

$$\alpha = \frac{1}{1 + \frac{K_1 K_p L}{T}} = \frac{1}{1 + \frac{L}{T_o}}$$

$$K_c = \frac{T}{(L + T_o)K_p}$$

Neste controlador, T_o é o parâmetro de ajuste que regula a velocidade de resposta em malha fechada e ao mesmo tempo a robustez.

Exemplo ilustrativo:

Considere o processo $P(s) = \frac{e^{-L_n s}}{(s+1)(0.5s+1)(0.2s+1)}$ para $L_1 = 0.5$, $L_2 = 8$ e seus respectivos modelos²:

$$P_{n1}(s) = \frac{e^{-s}}{1.26s+1}$$

$$P_{n2}(s) = \frac{e^{-8.5s}}{1.28s+1}$$

Na figura 2.5 são mostradas as respostas ao degrau aplicado em $t = 3$ e em malha fechada para L_1 e L_2 , onde o controlador é calculado com um $T_o = 0.4$. O efeito do atraso fica bastante claro: uma resposta mais oscilatória e com um tempo de acomodação maior. Uma solução seria aumentar o tempo de resposta em malha fechada, T_o , tornando o sistema mais lento, mas com uma oscilação menor. Na figura 2.6 são comparadas as respostas para $T_o = 0.4$ e para um novo ajuste, $T_o = 2.8$. Fica como análise do projetista definir as prioridades em função das especificações exigidas, ou seja, o que é mais importante para cada situação: uma resposta mais lenta com um sistema mais robusto ou um sistema mais rápido, porém com uma resposta mais oscilatória.

Para sistemas onde o atraso representa uma interferência maior na resposta do sistema, como visto neste exemplo, uma solução mais adequada seria o uso do DTC. Na figura 2.7, são comparadas as respostas do PS e do PID para $L_2 = 8$. Em ambos os casos, utiliza-se o mesmo modelo, P_{n2} , e o mesmo parâmetro de ajuste do controle, $T_o = 2.8$. Nota-se que o PS oferece um desempenho melhor ao sistema.

Na próxima seção, algumas abordagens dos DTCs serão analisadas, assim como suas implementações e alguns métodos de ajustes.

²Os modelos foram obtidos através da ferramenta desenvolvida neste trabalho e, para aproximar de situações reais, ruídos foram aplicados na identificação dos processos simulados.

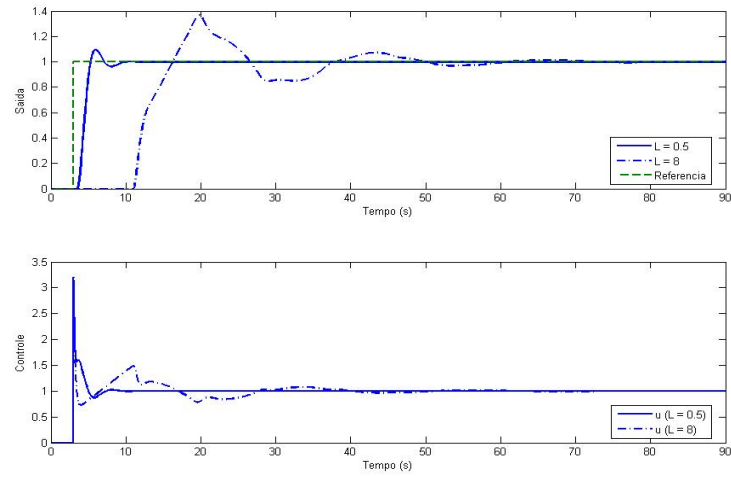


Figura 2.5: Respostas ao degrau em malha fechada para $L_1 = 0.5$, $L_2 = 8$ e $T_o = 0.4$

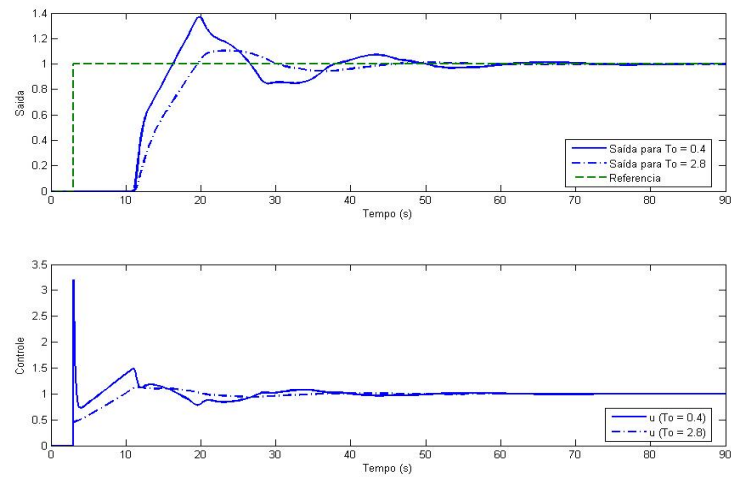


Figura 2.6: Respostas ao degrau em malha fechada para $T_o = 0.4$ e $T_o = 2.8$

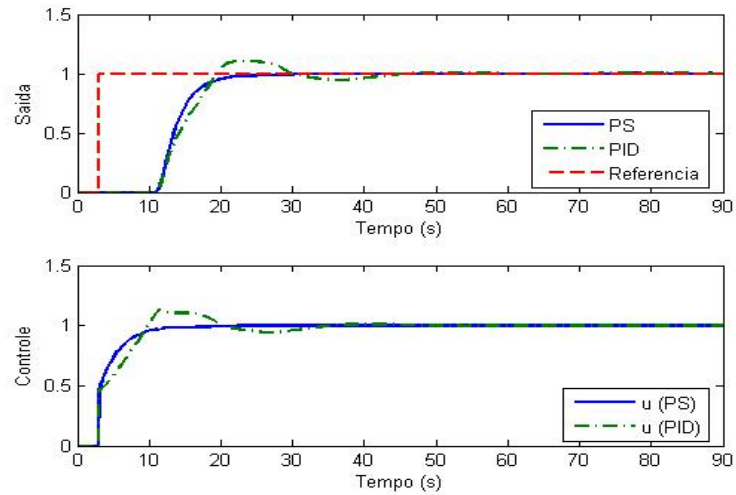


Figura 2.7: Respostas ao degrau em malha fechada do PS e do PID para $L = 8$

```

% processo 1
nump1=1;denp1=conv([1 1],conv([.2 1],[.5 1]));L1=0.5;
% modelo 1
T=1.26;Kp=1;Ln1=1;
numpn1=Kp;denpn1=[T 1];

% controle PID1
k1=3;Ti=T;Td=0.5*Ln1;
k1=T/(To*Kp);alpha=1/(1+Ln1/To);Kc=T/((Ln1+To)*Kp);
numcont=Kc*conv([Ti 1],[Td 1]);dencont=conv([Ti 0],[alpha*Td 1]);

% controle PS1
numc=k1*[Ti 1];
denc=[Ti 0];

sim('ex4')
plot(tout,yp,tout,yp,'-.',tout,ref,'--','linewidth',2)

```

2.4 Controladores DTC

Processos que contêm um atraso significativo podem ter seu desempenho em malha fechada melhorada por uma estrutura de predição. Controladores com estas estruturas são denominados DTCs. O primeiro e um dos mais conhecidos é o PS, apresentado na década de 50 para melhorar o desempenho de controladores clássicos (PI, PID) para processos com atraso. O seu algoritmo é o mais utilizado na indústria para a compensação do atraso.

Nos últimos 25 anos, diversas extensões e modificações do PS foram propostas no sentido de melhorar as rejeições de perturbações, permitir implementações em plantas instáveis, tornar o sistema mais robusto e facilitar o ajuste nas aplicações industriais.

É muito comum se pensar que os DTCs devam ser utilizados quando o processo apresenta um atraso dominante ($L \geq 2T$). No entanto, as estruturas dos DTCs são mais complexas e requerem um maior conhecimento de ajuste, por parte do operador, do que os tradicionais PIDs. Existem alguns estudos na literatura onde são analisadas implementações de PIDs e DTCs, comparando seus desempenhos e robustez [33], [18], [25]. Estes trabalhos tentam esclarecer uma dúvida dos usuários de controladores industriais: quando será vantajoso usar um DTC?

Em [33], o PS é comparado com o PID, controlando processos de primeira ordem com atraso e aplicando um degrau como referência ou perturbação. Mostra-se que o ISE (*Integrated Square Error*) pode ser melhorado em mais de 10%, se não for levada em conta a razão L/T . Ou seja, para valores pequenos de L/T onde o PID apresenta um resultado satisfatório, este melhoramento não é atingido devido às incertezas de modelagem que comprometem a ação do PS, fazendo com que não haja muita diferença entre o desempenho do PID e do PS. Para valores grandes de L/T , o uso do PS ao invés do PID se torna bastante vantajoso, se o modelo do processo for válido para uma largura de faixa suficiente no domínio da frequência.

Em [18], o IAE (*Integrated Absolute Error*) é comparado com controladores PI, PID e um simples DTC para processos estáveis e integrativos, quando um degrau é aplicado como perturbação e uma certa robustez é admitida. A conclusão da análise é que o DTC oferece um desempenho melhor que o do PI para $T/L \in [0.1, 5]$ e o do PID é melhor que o do DTC para $T/L \in [1, 10]$. Entretanto, estes resultados foram encontrados para valores fixos de D_M (margem do atraso³) e o DTC melhora o desempenho quando valores pequenos de D_M são considerados. Para processos integrativos, a diferença de desempenhos entre o PI e o DTC é bem maior. O DTC reduz muito mais o valor do IAE em relação ao PI.

Já em [25], é apresentada uma análise diferente. O PID é considerado como um PS com o modelo do atraso calculado pela aproximação de Padé, já mostrado na seção 2.3.1. A principal conclusão desta análise é que as vantagens do DTC são diretamente relacionadas ao erro de modelagem do atraso, independentemente do valor nominal de L . Geralmente, as vantagens do DTC são mais notáveis quando os erros de modelagem são pequenos e mais ainda se o atraso é dominante.

Em geral, se o atraso é pequeno, podendo ainda ser mal estimado, e o modelo é de baixa ordem, o PID permite a obtenção de uma resposta mais lenta e aceitável, com um ajuste conservador. Mas se o atraso é grande e razoavelmente bem estimado e deseja-se uma resposta mais rápida em malha fechada, é mais conveniente utilizar sistemas de compensação de atraso.

Muitas estruturas de DTC têm sido propostas na literatura. Do ponto de vista da ferramenta, interessa, neste trabalho, mostrar características gerais de análise e projeto destes métodos, quando considera-se seguimento de referência, rejeição de perturbações e robustez. Uma estrutura de DTC que permite isto é a do DO-DTC (*Disturbance Observer Dead-time Compensator*) que será analisada na continuação.

³Erro máximo admitido do atraso para manter o sistema estável em malha fechada

Ajuste do DO-DTC com dois graus de liberdade

O DO-DTC é um controlador 2DOF (*Two Degree of Freedom*) que utiliza a idéia de controle por pré-alimentação de perturbações. Em processos com perturbações mensuráveis, é interessante usar um controle antecipativo, já que esta estrutura permite diminuir o efeito das perturbações. Em casos de perturbações não mensuráveis, um observador pode ser projetado a partir das medidas de saída e de controle a fim de estimar uma perturbação.

Considera-se a estrutura ilustrada na figura 2.8, onde $Y(s) = P(s)(U(s) + Q(s))$, a planta é representada por $P(s) = G(s)e^{-Ls}$ e os controladores que serão projetados, $F(s)$ e $V(s)$:

Neste sistema, a perturbação pode ser calculada, idealmente, como:

$$Q(s) = P^{-1}(s)Y(s) - U(s) \quad (2.20)$$

Nota-se que o termo P^{-1} contém o termo e^{Ls} , portando a solução é não realizável. Para resolver este problema, a equação (2.20) é multiplicada por e^{-Ls} , obtendo-se:

$$e^{-Ls}Q(s) = G^{-1}(s)Y(s) - e^{-Ls}U(s) \quad (2.21)$$

Observa-se que, neste caso, a perturbação é obtida com um atraso L .

Usando um modelo, $P_n(s)$, e um filtro, $V(s)$, que torna o termo $V(s)G_n^{-1}(s)$ causal e estável, garantindo a implementação do controlador [25], uma perturbação com atraso estimada, $\hat{Q}(s)$, pode ser calculada:

$$\hat{Q}(s) = V(s)[G_n^{-1}(s)Y(s) - e^{-L_n s}U(s)] \quad (2.22)$$

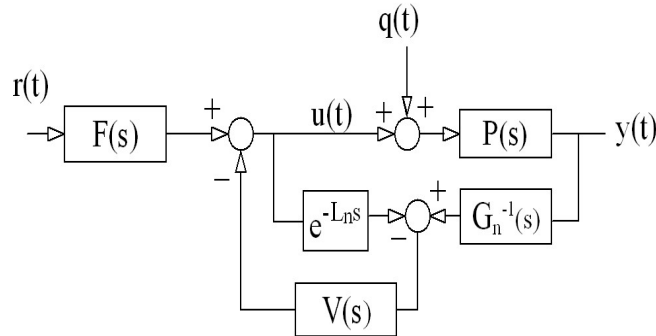


Figura 2.8: Estrutura para análise do DO-DTC

Neste sistema, a função de transferência nominal entre a referência e a saída é dada por:

$$\frac{Y(s)}{R(s)} = F(s)G_n(s)e^{-Ls} \quad (2.23)$$

e entre a perturbação e a saída:

$$\frac{Y(s)}{Q(s)} = P_n(s)(1 - V(s)e^{-Ls}) \quad (2.24)$$

Se forem considerados os erros na modelagem, $P(s) = P_n(s)(1 + \delta P(s))$, a equação característica é:

$$1 + V(s)e^{-Ls}\delta P(s) = 0$$

onde $\delta P(s)$ é o erro de modelagem multiplicativo.

Então, a robustez do sistema em malha fechada é dada por:

$$\overline{\delta P}(\omega) < dP(\omega) = \frac{1}{|V(j\omega)|} \quad \forall \omega \in [0, \infty[\quad (2.25)$$

Nota-se pelas equações (2.23), (2.24) e (2.25) que o controle desacopla as respostas à referência e à perturbação. A robustez do sistema é definida por $V(s)$ e este, por sua vez, afeta também a rejeição de perturbações. Desta forma, $V(s)$ é ajustado assumindo um compromisso entre robustez e rejeição de perturbações, enquanto $F(s)$ é ajustado para um comportamento de saída desejado às referências. Essa estrutura também mostra que o controlador tem, no mínimo, uma ação integral ($1 - V(s)e^{-Ls} = 0$ para $s = 0$). Então, o sistema em malha fechada rastreia mudanças do tipo degrau na referência.

Para rejeitar perturbações do tipo $\frac{1}{s^m}$ em regime permanente, o termo $P_n(s)[1 - V(s)e^{-Ls}]$ deve conter m raízes em $s = 0$ no seu numerador. Isto implica:

$$\left\{ \begin{array}{l} (1 - V(s)e^{-Ls})|_{s=0} = 0 \\ \frac{d}{ds}(1 - V(s)e^{-Ls})|_{s=0} = 0 \\ \vdots \\ \frac{d^{m-1}}{ds^{m-1}}(1 - V(s)e^{-Ls})|_{s=0} = 0 \end{array} \right. \quad (2.26)$$

onde m é a ordem da perturbação (1 para degraus, 2 para rampas, etc). Em caso de plantas integradoras, o termo $(1 - V(s)e^{-Ls})$ deve possuir um zero em $s = 0$ para cada integrador, compensando seu efeito. E em casos de plantas instáveis, o termo $(1 - V(s)e^{-Ls})$ deve

possuir um zero em cada pólo instável de $P_n(s)$.

A estrutura da figura 2.8 serve mais para análise, dado que não pode ser implementada diretamente. Para implementação, a estrutura da figura 2.9 deve ser usada, pois permite evitar instabilidade interna no caso de plantas integradoras ou instáveis.

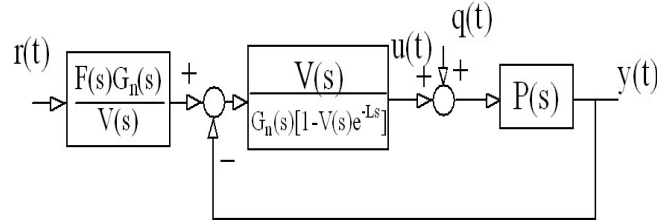


Figura 2.9: Estrutura para implementação do DO-DTC

Exemplo ilustrativo:

Considera-se um processo e seu modelo como:

$$P(s) = \frac{1.5e^{-15s}}{(2s+1)^5}$$

$$P_n(s) = \frac{1.5e^{-20.2s}}{5.7s+1}$$

O sistema em malha fechada deve seguir referência e também rejeitar perturbações do tipo rampa na entrada do processo. Uma rampa é aplicada em $t = 80s$. Para rejeitar rampas, $V(s)$ deve ter ordem 2. A forma geral de $V(s)$ é:

$$V(s) = \frac{\alpha_1 s + \alpha_0}{(T_v s + 1)^2}$$

onde α_1 e α_2 são calculados com a condição da equação (2.26) e T_v é o parâmetro de ajuste para atingir um compromisso entre desempenho e robustez. Neste caso, usa-se $T_v = 6.5$, obtendo-se:

$$\alpha_1 = 33.2; \alpha_0 = 1$$

No segundo passo, ajusta-se o filtro $F(s)$. É importante notar que, como o ganho de $V(s)$ é unitário e do modelo $P_n(s)$ é 1.5, para $s \rightarrow 0$, então o filtro $F(s)$ deve ter um ganho de $\frac{1}{1.5}$ para que o filtro equivalente, $F_{eq} = \frac{F(s)G_n(s)}{V(s)}$, tenha um ganho unitário e o sistema siga a

referência corretamente. Assim, neste caso, pode-se escolher:

$$F(s) = \frac{1}{1.5}$$

dado que não deseja-se acelerar mais a resposta.

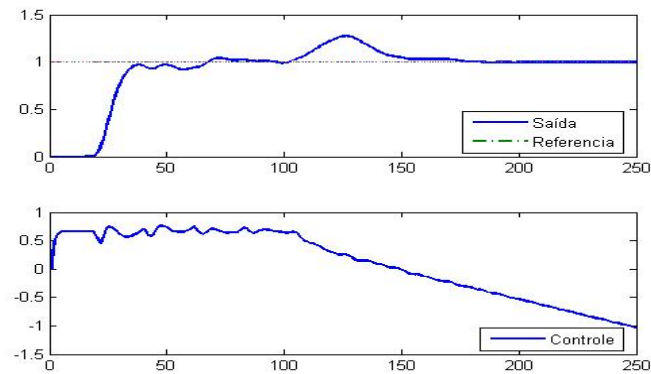


Figura 2.10: Resposta do sistema com uma perturbação do tipo rampa em $t=80s$

Código do MATLAB®

```
% processo
L=15;numG=1.5;denG=[32 80 80 40 10 1];

% modelo
Ln=20.2;numGn=1.5;denGn=[5.7 1];

% Controlador
Tv=6.5;alpha0=1;alpha1=33.2;numV=[alpha1 alpha0];
denV=[Tv 1];denV=conv(denV,denV);numF=1;denF=[1.5];

% Plotagem
sim('do-dtc')
subplot(2,1,1)
plot(tout,y,tout,ref,'linewidth',2)
legend('Saída','Referencia','location','southeast')
subplot(2,2,2)
plot(tout,u,'linewidth',2)
legend('Controle','location','southeast')
```

Discretização do DTC

Na época em que o PS foi proposto, ainda eram usados equipamentos de controle analógicos. A implementação destes equipamentos era muito difícil e os DTCs não foram utilizados até a década de 80, quando surgiram equipamentos digitais na indústria. A partir de então, a idéia de compensação de atraso foi estendida para os tradicionais processos contínuos, os quais tinham suas dinâmicas descritas no domínio do tempo discreto. Nestes casos, o projeto dos controladores era feito por ferramentas discretas.

Neste trabalho não foi diferente. Para a implementação de alguns controladores da ferramenta, foi necessário desenvolver alguns algoritmos que tratam diretamente com os sistemas discretos ou com as discretizações dos sistemas contínuos. É importante lembrar que o termo que representa o atraso contínuo é uma função irracional e analisar polinômios com termos irracionais não é uma tarefa muito simples. Em algumas situações, por exemplo, para garantir a estabilidade interna de um sistema, é necessário que as raízes comuns de um polinômio e um pseudo-polinômio sejam canceladas. Por exemplo, no controlador DO-DTC, a função $S(s)$ deve ser estável, mesmo $G_n(s)$ sendo instável:

$$S(s) = G_n(s)[1 - V(s)e^{-Ls}] = \frac{N_g(s)}{D_g(s)}[1 - V(s)e^{-Ls}] \quad (2.27)$$

Para tal, é preciso que exista cancelamento entre zeros de $(1 - V(s)e^{-Ls})$ e pólos instáveis de $G_n(s)$ na implementação do controlador. No caso contínuo, isto é muito complexo e inconveniente, podendo ser realizado com aproximações polinomiais do atraso ou usando implementações não dinâmicas [25]. No caso discreto, apenas é necessário dividir os polinômios resultantes.

O desempenho do controlador não somente depende da escolha do período de amostragem como também do método de discretização usado para transformar a equação diferencial original, do processo ou do controlador, em um modelo discreto [16].

De fato, as mais importantes características do DTC contínuo podem ser mantidas no caso discreto, se o período de amostragem for muito pequeno em relação à constante de tempo dominante do sistema. Entretanto, na prática, quando há limitações na implementação em tempo real, o período de amostragem pode ter que ser escolhido de forma que seu valor afete o desempenho e a robustez do sistema em malha fechada, especialmente nos sistemas com atraso.

Para a análise das propriedades de um DTC discreto, uma representação unificada do

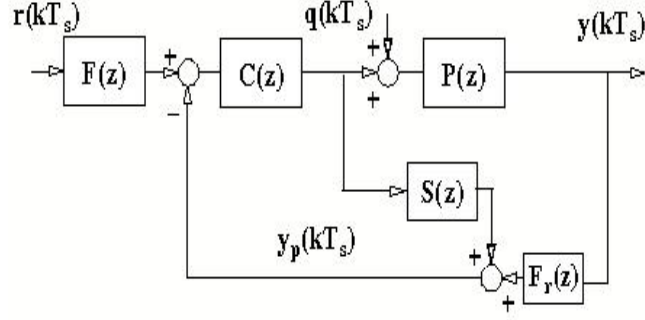


Figura 2.11: Representação unificada de DTCs discretos

DTC pode ser utilizada [25]. Considere o diagrama de blocos apresentado na figura 2.11, onde os blocos $F_r(z)$ e $S(z)$ representam a parte preditiva e os blocos $C(z)$ e $F(z)$ representam o controlador primário de dois graus de liberdade. Esta estrutura é denominada UDTC (Unified Dead-time Compensator), onde as funções de transferência entre a referência, perturbação e a saída são dadas por:

$$H_r(z) = \frac{F(z)C(z)P(z)}{1 + C(z)[P(z)F_r(z) + S(z)]} \quad (2.28)$$

$$H_q = \frac{P(z)[1 + C(z)S(z)]}{1 + C(z)[P(z)F_r(z) + S(z)]} \quad (2.29)$$

Como no caso contínuo, a robustez é analisada, usando o erro de modelagem multiplicativo representado no domínio da frequência $\forall \omega \in [0, \pi/T_s]$:

$$\begin{aligned} \delta P(j\omega) &= \frac{P(e^{j\omega}) - P_n(e^{j\omega})}{P_n(e^{j\omega})} \leq \bar{\delta P}(\omega), \\ \bar{\delta P}(\omega) &< dP(j\omega) = \frac{|1 + C(e^{j\omega})[P_n(e^{j\omega})F_r(e^{j\omega}) + S(e^{j\omega})]|}{|G_n(e^{j\omega})C(e^{j\omega})F_r(e^{j\omega})|} \end{aligned} \quad (2.30)$$

Observa-se que $F_r(z)$ afeta $H_q(z)$ e $F(z)$ afeta somente $H_r(z)$.

Se o modelo discreto for obtido através de uma discretização do sistema contínuo, deve-se levar em consideração a interferência do amostrador e sustentador no sistema. Como o sinal de controle é aplicado no processo através de um segurador de ordem zero (ZOH), então o melhor método para se obter o modelo discreto do processo é a discretização baseada no ZOH. Desta forma, obtém-se a seguinte relação:

$$P^s(j\omega) = H_0(s)P(j\omega) \quad \forall \omega \in [0, \pi/T_s]$$

onde o termo $H_0(s) = \frac{1-e^{-j\omega T_s}}{j\omega T_s}$ representa a função de transferência do segurador de ordem zero.

Nota-se agora que com a implementação discreta, pode-se garantir a estabilidade interna do sistema, cancelando diretamente as raízes dos polinômios discretizados da equação (2.27), no caso de plantas instáveis:

$$S(z) = Z\{L^{-1}\{H_0(s)S(s)\}\} = \frac{N_g(z)}{D_g(z)}[1 - V(z)z^{-d}] \quad (2.31)$$

onde $Z\{L^{-1}\{Y(s)\}\}$ é a transformada Z da resposta ao impulso amostrada de $Y(s)$.

2.5 Controladores MPC

O MPC é uma das técnicas de controle moderno mais potentes e, provavelmente, a que teve maior êxito em aplicações industriais [44], principalmente, porque pode ser utilizado tanto em sistemas monovariáveis como multivariáveis, lineares ou não lineares e porque as restrições nas saídas e nas ações de controle podem ser consideradas no projeto da lei de controle em tempo real [6].

O MPC surge na indústria de processos no final dos anos 70. Em [40, 41] foi apresentado o MPHC (*Model Predictive Heuristic Control*), posteriormente conhecido como MAC (*Model Algorithm Control*) e em [32], o DMC (*Dynamic Matrix Control*). Um modelo da dinâmica do processo, é explicitamente utilizado no sentido de estimar os efeitos das ações de controle futuras que são determinadas pela minimização do erro predito entre a referência e a saída, sujeito às restrições operacionais. A otimização é feita em cada período de amostragem, atualizando os dados de informação do processo. Para implementar a estratégia do MPC, uma estrutura básica ilustrada na figura 2.12 é usada.

O MPC se tornou popular de forma rápida, sobretudo em processos químicos industriais, e vem se desenvolvendo consideravelmente nos últimos anos, tanto na área de pesquisa quanto na indústria. A razão para isto pode ser atribuída ao fato do MPC ser a forma mais geral de tratar o problema de controle de processos no domínio do tempo. A formulação do MPC integra controle ótimo, controle estocástico, controle monovariável e multivariável, controle linear e não linear e controle de processos com atraso. Além disto, com este tipo de controlador, a robustez frente aos erros de modelagem e ruído de medição pode ser considerada no algoritmo através do uso de polinômios de filtragem [9]. Por outro lado, o MPC ainda não

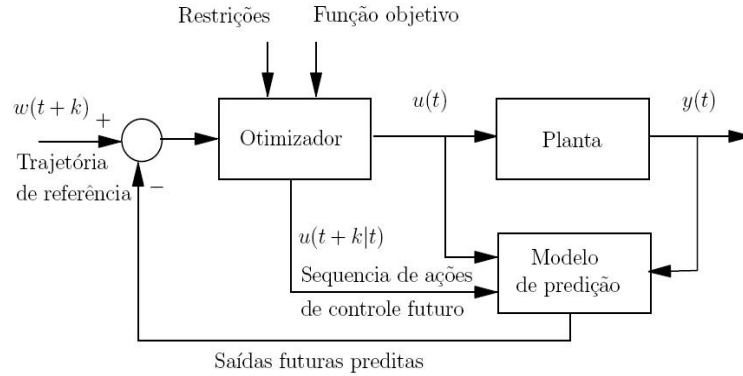


Figura 2.12: Estrutura Geral do MPC

atingiu um número de aplicações sugerido pelo seu potencial. A complexidade matemática existente na sua implementação prática, muitas vezes representa uma desvantagem para engenheiros de controle, mas que não significa um problema para a comunidade científica, haja vista diversos pacotes matemáticos encontram-se completamente disponíveis.

Os vários algoritmos do MPC apenas se diferem entre si pelo modelo usado para representar o processo, ruído e a função custo a ser minimizada. Atualmente, suas aplicações são diversas. Desde processos que contêm manipuladores robóticos [1] até anestesia clínica [2],[45]. Aplicações em indústrias de cimento, torres de secagem e braços robóticos são descritos em [7], enquanto desenvolvimentos para colunas de destilação, plantas PVC e máquinas a vapor são apresentados em [39, 41]. O bom desempenho destas aplicações, mostram o quanto o MPC é eficiente para controlar sistemas vulneráveis a intervenções por um longo período de tempo.

Apesar de ter uma formulação diferente, é possível demonstrar que a lei de controle do controlador preditivo sem restrições é equivalente à de um DTC discreto [25]. Isto permite a análise das propriedades destes controladores usando o conhecimento dos DTCs e ao mesmo tempo aproveitá-las para ajustar o MPC [25]. Assim, é importante que uma ferramenta de simulação permita integrar os controladores DTC e MPC.

Nesta seção, apresenta-se uma revisão de algumas estratégias de controle preditivo de sistemas lineares, utilizadas como base no desenvolvimento de parte deste trabalho: o GPC e o DTC-GPC (*Dead-time Compensator Generalized Predictive Control*).

2.5.1 O algoritmo GPC

O algoritmo GPC usa o modelo CARIMA (*Controlled Auto-Regressive Integrated Moving Average*) para calcular as previsões ao longo de um horizonte futuro [9]. O modelo CARIMA, para o caso monovariável, tem a seguinte forma:

$$A(z^{-1})y(t) = z^{-d}B(z^{-1})u(t-1) + \frac{T(z^{-1})e(t)}{\Delta} \quad (2.32)$$

onde $u(t)$ e $y(t)$ são o controle e saída da planta, $e(t)$ é ruído branco e d é o atraso. $A(z^{-1})$, $B(z^{-1})$, $T(z^{-1})$ e Δ são polinômios em função do operador atraso z^{-1} :

$$A(z^{-1}) = 1 + a_1z^{-1} + a_2z^{-2} + \dots + a_{n_a}z^{-n_a}$$

$$B(z^{-1}) = b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_{n_b}z^{-n_b}$$

$$T(z^{-1}) = 1 + t_1z^{-1} + t_2z^{-2} + \dots + t_{n_t}z^{-n_t}$$

$$\Delta = 1 - z^{-1}$$

Neste modelo, o polinômio $T(z^{-1})$ representa as características estocásticas do ruído e das perturbações. O principal objetivo do uso de $T(z^{-1})$ é reduzir os efeitos do ruído e das perturbações na predição de saída sem afetar o comportamento nominal. Por outro lado, as perturbações de baixa frequência podem ser reduzidas pelo operador Δ que aparece no modelo de predição.

Entretanto, nas aplicações industriais é difícil estimar as características do ruído. Portanto, $T(z^{-1})$ é raramente estimado, e sim escolhido como um filtro [9], [8], [42], [43], [30]. Se este é apropriadamente escolhido, o erro de predição pode diminuir devido ao erro de modelagem, que é particularmente importante em altas frequências. Nota-se que as incertezas devidas à má estimação do atraso são uma das características mais frequentes de dinâmica não modelada em altas frequências e tem uma influência negativa na estabilidade em malha fechada, uma vez que diminui a margem de fase do sistema.

O algoritmo GPC [9], consiste em aplicar uma sequência de controle que minimiza a seguinte função custo:

$$J = \sum_{k=N_1}^{N_2} [\hat{y}(t+k|t) - w(t+k)]^2 + \lambda \sum_{k=0}^{N_u-1} [\Delta u(t+k)]^2 \quad (2.33)$$

em que \hat{y} é a predição da saída do processo, Δu é a variação do sinal de controle, w é a

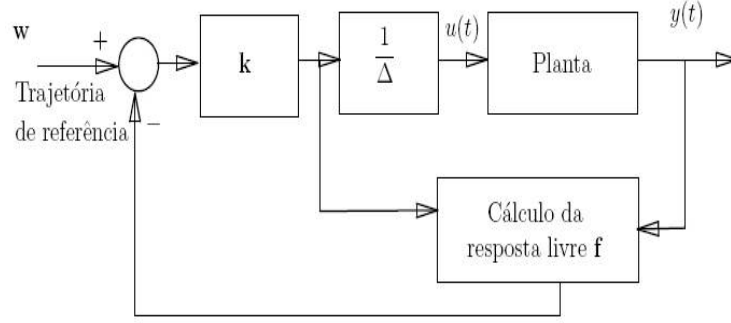


Figura 2.13: Estratégia de controle preditivo GPC

trajetória de referência futura, λ é a ponderação do controle, N_u é o horizonte de controle, N_1 e N_2 definem o horizonte de predição. Normalmente, N_1 e N_2 são dados por $d+1$ e $d+N$, em que d é o atraso do sistema e N a largura do horizonte de predição.

O valor da predição ótima em, $t+j$, pode ser calculada por meio da *equação Diophantina*⁴ [6]:

$$T(z^{-1}) = E_j(z^{-1})\Delta A(z^{-1}) + z^{-j}F_j(z^{-1}) \quad (2.34)$$

Usando esta equação e o modelo da planta (2.32), a saída futura da planta pode ser expressa por:

$$y(t+j) = \frac{F_j(z^{-1})}{T(z^{-1})}y(t) + \frac{E_j(z^{-1})B(z^{-1})}{T(z^{-1})}z^{-d}\Delta u(t+j-1) + E_j(z^{-1})e(t+j) \quad (2.35)$$

A expressão $E_j(z^{-1})e(t+j)$ da Equação (2.35) contém somente o ruído branco futuro, dado que o grau de $\{E_j(z^{-1})\} = (j-1)$. Considerando que a melhor estimativa deste ruído é zero, a predição ótima (denotada por \hat{y}) é dada por [6]:

$$\hat{y}(t+j|t) = \frac{F_j(z^{-1})}{T(z^{-1})}y(t) + \frac{E_j(z^{-1})B(z^{-1})}{T(z^{-1})}z^{-d}\Delta u(t+j-1) \quad (2.36)$$

Esta expressão está em função de valores passados e futuros das ações de controle. As ações de controle passadas podem ser separadas das futuras usando, neste caso, a seguinte equação Diophantina:

$$E_j(z^{-1})B(z^{-1})z^{-d} = H_j^*(z^{-1})T(z^{-1}) + z^{-j}I_j^*(z^{-1}) \quad (2.37)$$

⁴A *equação Diophantina* é denominada uma equação polinomial do tipo $P1(z^{-1}) = P2_j(z^{-1})P3(z^{-1}) + z^{-j}P4_j(z^{-1})$, em que $P2_j(z^{-1})$ e $P4_j(z^{-1})$ podem ser obtidas dividindo $P1(z^{-1})$ por $P3(z^{-1})$ até que o resto possa ser fatorado como $z^{-j}P4_j(z^{-1})$. O quociente será $P2_j(z^{-1})$.

Com isto, a predição da saída pode ser escrita como:

$$\begin{aligned}\hat{y}(t+j|t) &= H_j^*(z^{-1})\Delta u(t+j-1) + \frac{F_j(z^{-1})}{T(z^{-1})}y(t) + \frac{I_j^*(z^{-1})}{T(z^{-1})}\Delta u(t-1) \\ &= H_j^*(z^{-1})\Delta u(t+j-1) + F_j(z^{-1})y^f(t) + I_j^*(z^{-1})\Delta u^f(t-1)\end{aligned}\quad (2.38)$$

em que $y^f(t) = \frac{y(t)}{T(z^{-1})}$ e $\Delta u^f(t-1) = \frac{\Delta u(t-1)}{T(z^{-1})}$. Nota-se que j deve variar de $t+d+1$ a $t+d+N$. Por outro lado, os coeficientes nulos, devido aos atrasos de $H_j^*(z^{-1})$ e $I_j^*(z^{-1})$, podem ser eliminados, tornando-se $H_j(z^{-1})$ e $I_j(z^{-1})$. Assim, a predição da saída é dada por:

$$\hat{y}(t+j|t) = H_j(z^{-1})\Delta u(t+j-d-1) + F_j(z^{-1})\hat{y}^f(t) + I_j(z^{-1})\Delta u^f(t-1) \quad (2.39)$$

e sua representação vetorial por:

$$\hat{\mathbf{y}} = \mathbf{G}\Delta\mathbf{u} + \mathbf{F}\mathbf{y}_1^f + \mathbf{I}\mathbf{u}_1^f \quad (2.40)$$

onde:

$$\begin{aligned}\hat{\mathbf{y}} &= \begin{bmatrix} \hat{y}(t+d+1|t) \\ \hat{y}(t+d+2|t) \\ \vdots \\ \hat{y}(t+d+N|t) \end{bmatrix}, \quad \Delta\mathbf{u} = \begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \vdots \\ \Delta u(t+Nu-1) \end{bmatrix}, \\ \mathbf{y}_1^f &= \begin{bmatrix} \hat{y}(t) \\ \hat{y}(t-1) \\ \vdots \\ \hat{y}(t-na) \end{bmatrix}, \quad \mathbf{u}_1^f = \begin{bmatrix} \Delta u^f(t-1) \\ \Delta u^f(t-2) \\ \vdots \\ \Delta u^f(t-ni) \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} h_1 & 0 & \dots & 0 \\ h_2 & h_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \dots \\ h_N & h_{N-1} & \dots & h_{N-N_u+1} \end{bmatrix}, \\ \mathbf{F} &= \begin{bmatrix} f_{1+d,1} & f_{1,2} & \dots & f_{1+d,n_a+1} \\ f_{2+d,1} & f_{2,2} & \dots & f_{2+d,n_a+1} \\ \vdots & \vdots & \ddots & \vdots \\ f_{N+d,1} & f_{N,2} & \dots & f_{N+d,n_a+1} \end{bmatrix}, \quad \mathbf{I} = \begin{bmatrix} i_{1,1} & i_{1,2} & \dots & i_{1,n_i} \\ i_{2,1} & i_{2,2} & \dots & i_{2,n_i} \\ \vdots & \vdots & \ddots & \vdots \\ i_{N,1} & i_{N,2} & \dots & i_{N,n_i} \end{bmatrix}, \quad n_i = \max(n_b, n_t).\end{aligned}$$

Observa-se que os dois últimos termos da Equação (2.40) dependem somente de medidas passadas e podem ser agrupadas no vetor \mathbf{f} , denominado resposta livre do sistema. Portanto, a predição pode ser escrita como:

$$\hat{\mathbf{y}} = \mathbf{G}\Delta\mathbf{u} + \mathbf{f} \quad (2.41)$$

Uma vez conhecida a predição, a função custo (2.33) pode ser expressa como:

$$J = (\mathbf{G}\Delta\mathbf{u} + \mathbf{f} - \mathbf{w})^T(\mathbf{G}\Delta\mathbf{u} + \mathbf{f} - \mathbf{w}) + \Delta\mathbf{u}^T\mathbf{Q}_\lambda\Delta\mathbf{u} \quad (2.42)$$

em que $\mathbf{Q}_\lambda = \text{diag}\{\lambda_i\}$.

O valor mínimo de J (2.42), assumindo que não há restrições nos sinais de controle, pode ser encontrado igualando o gradiente de J a zero, o qual resulta na seqüência de controle ótima,

$$\Delta\mathbf{u} = (\mathbf{G}^T\mathbf{G} + \mathbf{Q}_\lambda)^{-1}\mathbf{G}^T(\mathbf{w} - \mathbf{f}) \quad (2.43)$$

Devido a estratégia de controle deslizante [9], o controle aplicado ao processo refere-se somente ao primeiro elemento de $\Delta\mathbf{u}$, isto é:

$$\Delta u(t) = \mathbf{k}(\mathbf{w} - \mathbf{f})$$

onde $\mathbf{k} = [k_1, k_2, \dots, k_N]$ (ver figura 2.13). No próximo período de amostragem, todo o processo de cálculo deve ser repetido considerando as novas medidas da saída do processo.

Uma propriedade importante é que a lei de controle do MPC para sistemas lineares com atraso pode ser definida e implementada na estrutura 2DOF-DTC, onde o controlador primário é calculado, utilizando um procedimento ótimo. Na figura 2.14 é mostrada a estrutura do DTC equivalente para o GPC. A ordem das funções de transferência que definem a lei de controle $W(z)$, $C(z)$ e $R(z)$ depende do grau dos polinômios do modelo de predição e o valor dos seus coeficientes depende dos parâmetros de ajuste do controlador, N_u , N , λ e $T(z^{-1})$ [25].

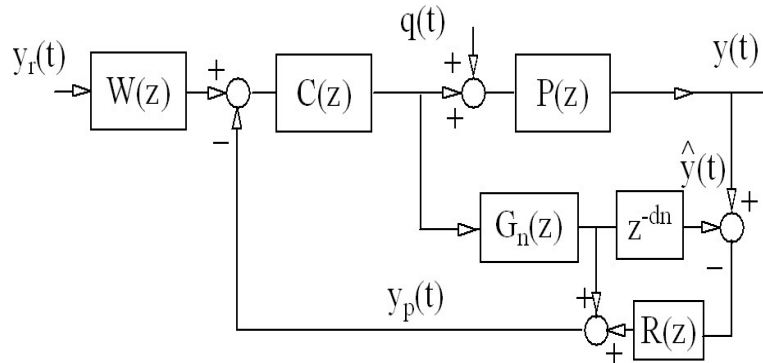


Figura 2.14: Diagrama de blocos do GPC

Nesta estratégia de controle, os parâmetros N_u , N e λ definem o desempenho de malha

fechada no caso nominal e o polinômio $T(z^{-1})$ pode ser ajustado para melhorar a robustez. Entretanto, o ajuste do polinômio $T(z^{-1})$ não é tão simples. Embora existam ajustes para alguns casos específicos, não há um procedimento sistemático que possa ser usado de forma geral [6]. Por outro lado, sabe-se que o GPC pode ser aplicado em processos integradores ou instáveis com atraso.

Como a estrutura do GPC é equivalente à do DTC, é possível realizar uma modificação no cálculo das previsões para obter um GPC apropriado para sistemas com atraso.

Em seguida, será mostrado um compensador de atraso baseado no GPC no qual é possível melhorar a robustez do sistema de forma simples e eficiente, através do uso de um filtro.

2.5.2 O Compensador de atraso baseado no GPC

O Compensador de atraso baseado no GPC (DTC-GPC) utiliza o mesmo procedimento de otimização do GPC, mas calcula as previsões de forma diferente. Estudos realizados em [26], mostram que para processos lineares, estáveis e com atraso, as propriedades do GPC podem ser bastante melhoradas, utilizando um preditor de Smith, no lugar do preditor ótimo, para calcular as previsões até o instante $t + d$. Com isto, pode-se obter o mesmo desempenho nominal e melhor robustez que o GPC, principalmente, quando existem erros ao estimar o atraso do processo.

No DTC-GPC, a previsão da saída é calculada em duas etapas. Na primeira é calculada a previsão da saída até o instante $t + d$ e na segunda, no intervalo $[t + d + 1, \dots, t + d + N]$. Para calcular as previsões até o tempo $t + d$, utiliza-se o modelo da planta em malha aberta sem considerar perturbações e corrige-se a previsão, usando-se as diferenças entre a previsão calculada e a saída real: $\hat{y}(t + d - i | t) \leftarrow \hat{y}(t + d - i | t) + y(t - i) - \hat{y}(t - i)$. Este erro de previsão pode ser filtrado antes de ser somado ao valor da previsão em malha aberta, o que permite melhorar a robustez do preditor de Smith em caso de sistemas com incertezas no atraso [29]. Com este procedimento, a previsão no instante $t + d$ é dada por:

$$\hat{y}(t + d | t) = G(z)u(t) + R(z)[y(t) - P(z)u(t)] \quad (2.44)$$

onde $R(z)$ é um filtro passa baixa que deve ser ajustado junto com os outros parâmetros do

controlador [26] e P e G representam o modelo da planta:

$$P(z) = \frac{B(z^{-1})z^{-1-d}}{A(z^{-1})} = G(z)z^{-d}$$

A predição a partir do instante $t+d+1$ é calculada, considerando o modelo de predição da Equação (2.32) com $T(z^{-1}) = 1$ [9] e usando a equação Diophantina $1 = E_j(z^{-1})\tilde{A}(z^{-1}) + F_j(z^{-1})z^{-j}$:

$$\hat{y}(t+d+j|t) = G_j(z^{-1})\Delta u(t+j-1) + F_j(z^{-1})\hat{y}^f(t+d|t) \quad (2.45)$$

em que $G_j(z^{-1}) = E_j(z^{-1})B(z^{-1})$. Separando as ações de controle futuras das ações passadas e colocando-se da forma vetorial, obtém-se:

$$\hat{\mathbf{y}} = \mathbf{G}\Delta\mathbf{u} + \mathbf{f} = \mathbf{G}\Delta\mathbf{u} + \mathbf{G}_p\mathbf{u}_1 + \mathbf{F}\mathbf{y}_1 \quad (2.46)$$

onde:

$$\begin{aligned} \hat{\mathbf{y}} &= \begin{bmatrix} \hat{y}(t+d+1|t) \\ \hat{y}(t+d+2|t) \\ \vdots \\ \hat{y}(t+d+N|t) \end{bmatrix}, \quad \Delta\mathbf{u} = \begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \vdots \\ \Delta u(t+N-1) \end{bmatrix}, \\ \mathbf{y}_1 &= \begin{bmatrix} \hat{y}(t+d|t) \\ \hat{y}(t+d-1|t) \\ \vdots \\ \hat{y}(t+d-na|t) \end{bmatrix}, \quad \mathbf{u}_1 = \begin{bmatrix} \Delta u(t-1) \\ \Delta u(t-2) \\ \vdots \\ \Delta u(t-nb) \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} g_1 & 0 & \dots & 0 \\ g_2 & g_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \dots \\ g_N & g_{N-1} & \dots & g_{N-N_u+1} \end{bmatrix}, \\ \mathbf{F} &= \begin{bmatrix} f_{1,1} & f_{1,2} & \dots & f_{1,n_a+1} \\ f_{2,1} & f_{2,2} & \dots & f_{2,n_a+1} \\ \vdots & \vdots & \ddots & \vdots \\ f_{N,1} & f_{N,2} & \dots & f_{N,n_a+1} \end{bmatrix}, \quad \mathbf{G}_p = \begin{bmatrix} g'_{1,1} & g'_{1,2} & \dots & g'_{1,n_b} \\ g'_{2,1} & g'_{2,2} & \dots & g'_{2,n_b} \\ \vdots & \vdots & \ddots & \vdots \\ g'_{N,1} & g'_{N,2} & \dots & g'_{N,n_b} \end{bmatrix}. \end{aligned}$$

Substituindo o valor de $\hat{\mathbf{y}}$ na função custo e minimizando-a, calcula-se o controle ótimo de forma igual ao item anterior. Neste caso, o controle ótimo é dado por [26]:

$$\Delta\mathbf{u} = (\mathbf{G}^T\mathbf{G} + \mathbf{Q}_\lambda)^{-1}\mathbf{G}^T(\mathbf{w} - \mathbf{f}) \quad (2.47)$$

Nota-se que \mathbf{f} está em função da predição até $t+d$ que é calculada pela Equação (2.44).

Observa-se que as soluções de controle do GPC e DTC-GPC, estruturalmente, são

iguais (Equações 2.43 e 2.47). A única diferença está no cálculo da resposta livre \mathbf{f} . No caso do GPC, a resposta livre \mathbf{f} é calculada com base em um preditor ótimo e no caso do DTC-GPC, a resposta livre \mathbf{f} é calculada com base em um preditor robusto (até $t + d$). Assim, este algoritmo usa as idéias dos DTC para a compensação do atraso e o ajuste da robustez e os critérios de resposta ótima do MPC para calcular o controle primário da estrutura. Desta forma, combina as vantagens das técnicas estudadas.

O ajuste do DTC-GPC segue as mesmas idéias do GPC no que se refere aos parâmetros λ , N_u e N , mas adiciona o parâmetro $R(z)$ como elemento para melhorar a robustez para controlar sistemas com atraso.

Ajuste do filtro $R(z)$

Considerando os erros de modelagem e calculando a função de transferência em malha fechada entre a entrada, a perturbação e a saída da estrutura da figura 2.14, obtém-se:

$$\frac{Y(z)}{Y_r(z)} = \frac{F(z)C(z)P(z)}{1 + C(z)[P(z)R(z) - P_n(z)R(z) + G_n(z)]} \quad (2.48)$$

$$\frac{Y(z)}{Q(z)} = P(z) \left[\frac{1 + C(z)G_n(z) - C(z)R(z)P_n(z)}{1 + C(z)G_n(z) - C(z)R(z)P_n(z) + C(z)R(z)P(z)} \right] \quad (2.49)$$

onde $P(z) = P_n(z)[1 + \delta P(z)]$ e sua equação característica é:

$$1 + C(z)P_n(z)\delta P(z)R(z) + C(z)G_n(z)$$

Assim, a condição geral de estabilidade robusta é [37]:

$$|\delta P(z)| < dP(z) = \frac{|1 + C(z)G_n(z)|}{|C(z)P_n(z)R(z)|} \quad z = e^{jw} \quad \forall w \in [0, \pi] \quad (2.50)$$

onde $dP(z)$ é o índice de robustez.

Desta equação, nota-se que o índice de robustez pode ser melhorado diminuindo o módulo do filtro e, da equação (2.49), que o filtro afeta também a rejeição de perturbações.

No caso nominal, tem-se:

$$\frac{Y(z)}{Q(z)} = P_n(z) \left[1 - \frac{C(z)R(z)P_n(z)}{1 + C(z)G_n(z)} \right] \quad (2.51)$$

O módulo à direita do segundo termo da equação (2.51) é o inverso do índice de robustez, $dP(z)$. Ou seja, melhorando o desempenho na rejeição de perturbações, o índice de robustez é reduzido. E para garantir a rejeição da perturbação e a estabilidade interna, $R(z)$ deve verificar condições equivalentes às do filtro $V(s)$. No caso do DO-DTC:

$$\left\{ \begin{array}{l} (R(z) - z^{dn})|_{z=z_0} = 0 \\ (R(z) - z^{dn})|_{z=1} = 0 \\ \frac{d}{dz}(R(z) - z^{dn})|_{z=1} = 0 \\ \vdots \\ \frac{d^{m-1}}{dz^{m-1}}(R(z) - z^{dn})|_{z=1} = 0 \end{array} \right. \quad (2.52)$$

onde $m = m_1 + m_2$, m_1 é a ordem da perturbação (1 para degraus, 2 para rampas...) e m_2 é a ação integradora da planta (0 para plantas estáveis, 1 para planta com um integrador, 2 para planta com duplo integrador...) e z_0 são as raízes “instáveis” da planta.

A implementação de $R(z)$ deve garantir que $\frac{R(z)}{G_n(z)}$ seja causal e, finalmente, a lei de controle se executa usando as idéias de resposta livre e forçada.

Capítulo 3

A Ferramenta

O Matlab (*The Mathworks, Inc*) é uma ferramenta poderosa e altamente otimizada para a realização de diversos procedimentos de processamento matemático. Especialmente adequado para áreas como engenharia, física e economia, este aplicativo dispõe de rotinas pré-programadas que representam economia significativa de tempo na resolução de problemas. Pesquisadores acadêmicos nos grandes centros universitários e empresas de tecnologia estão, cada vez mais, adotando o Matlab como ambiente de programação [19].

Apesar dos usuários do Matlab realizarem suas tarefas usando um sistema operacional com recursos gráficos, uma parcela significativa destes programadores não utiliza uma interface gráfica em seus códigos. O foco tradicional dos programadores de ambiente Matlab situa-se no processamento matemático apenas para a resolução de problemas avançados.

Desta forma, a criação de ferramentas de auxílio ao engenheiro baseadas no processamento matemático de Matlab e com uma interface gráfica agradável, com botões intuitivos e todos os outros recursos tradicionais de programas do ambiente do sistema operacional utilizado, é de grande interesse. Neste trabalho, estas características também foram consideradas. Nas próximas seções, o software será apresentado assim como serão analisadas, de forma geral, sua estrutura, navegação e aplicações.

3.1 Estrutura

A ferramenta foi desenvolvida em ambiente GUIDE (*Graphic User Interface Design Environment*) do Matlab, mantendo todos os seus formatos (números, vetores, matrizes,



Figura 3.1: Interface principal

funções, etc) e facilitando a comunicação com o usuário. O pacote Simulink foi utilizado para a geração de sinais e implementações de algoritmos de controle e identificação. Nesta primeira versão, a ferramenta está direcionada aos sistemas monovariáveis, tendo como algumas características:

- os dados de entrada e saída são fornecidos através da interface gráfica de forma simples e rápida;
- para diferentes tipos de simulação, não é necessário abrir novas janelas principais;
- janelas de “erro” e de “atenção” procuram orientar o bom uso por parte do usuário;
- a ferramenta possui uma memória onde o usuário pode salvar resultados, auxiliando nas comparações das diferentes simulações;
- legendas sugestivas são geradas pela própria ferramenta. No entanto, o usuário tem a liberdade de alterá-las;
- os algoritmos foram desenvolvidos usando como base os métodos apresentados no livro *Control of Dead-time Processes* [25];
- o usuário não precisa fazer readaptações para a implementação, tais como algoritmos e ordem de vetores e matrizes, muito comuns em simulações no Matlab. Tudo é calculado internamente pela ferramenta;

- não é utilizado o Workspace para o armazenamento de variáveis.

O sistema ainda possui uma interface inicial onde o usuário se comunica com os dois grupos de ferramentas: o de identificação e o de simulação, análise e projeto de controladores (ver figura 3.1). Em *Resultados*, o usuário poderá fazer análise e comparações dos seus projetos. Isto será visto mais adiante.

3.1.1 Aspectos de implementação

O fato da ferramenta permitir a interação do usuário com diversas estruturas de identificação, controle e comparação de forma fácil e rápida, tornou-se necessário o desenvolvimento de algoritmos que garantissem tanto a comunicação com todas as partes da ferramenta, assim como as implementações corretas referentes à teoria de sistemas de controle.

As leis básicas de controle como a do PID e a do PS foram implementadas sem a necessidade de uma programação mais sofisticada. Alguns comandos já próprios do Matlab e blocos de geração de sinais do Simulink foram suficientes para atender as diversas situações de análise. Porém, para os controladores mais avançados, diversos algoritmos foram necessários. O resultado obtido e a dimensão da ferramenta justificam a quantidade de algoritmos criada.

É importante frisar que as resoluções das equações e os requisitos de implementações apresentados no capítulo 2 são alguns dos objetivos alcançados, já que neste documento não foram apresentadas todas as teorias de controle e identificação correspondentes ao trabalho.

A implementação analógica dos DTCs contínuos é difícil devido ao fato destes controladores incluírem atrasos em sua estrutura. Isto tem motivado o uso de controladores digitais no controle de sistemas com atraso. Desta forma, os controladores DO-DTC e o Preditor de Smith Filtrado, desenvolvidos neste trabalho, em caso de sistema instável, só podem ser usados em tempo discreto.

Na ferramenta de identificação, no sentido de aproximar situações reais, os dados do processo são simulados em tempo discreto, embora o usuário obtenha os modelos discreto e contínuo fornecidos pela ferramenta. Isto será mostrado na próxima seção.

Implementação do DO-DTC

No capítulo 2 foram apresentadas todas as condições de implementação do DO-DTC. Mostrou-se que tanto para modelos instáveis como para modelos de fase não-mínima (zeros

no semi-plano direito no domínio s), é necessário calcular um filtro $V(s)$ que garanta a estabilidade interna e a realização do sistema; e um filtro $F(s)$ que determine o filtro equivalente $F_{eq}(z)$ com um ganho unitário, no caso de uma implementação discreta.

Nesta versão da ferramenta, o usuário fica restrito às análises de modelos de 1ª ou 2ª ordem, estáveis ou instáveis e/ou de fase não-mínima. Este último para os modelos de 2ª ordem, já que não faz muito sentido, na prática, um modelo de 1ª ordem com zero. Assim, o usuário ocupa-se apenas no ajuste de um parâmetro do filtro $V(s)$ em função da robustez e rejeição de perturbação, optando por uma implementação contínua ou discreta

Implementação discreta do PS filtrado para plantas integradoras e instáveis

Neste trabalho, a implementação do PS filtrado foi feita no tempo discreto, pois a forma exata no domínio contínuo é muito complexa para processos instáveis, dado que o controlador é internamente instável [26]. Entretanto, o usuário pode ajustar os parâmetros tanto no tempo contínuo quanto diretamente no tempo discreto.

Os parâmetros do filtro de robustez são calculados em função do parâmetro α , ajustado pelo usuário. Caso não seja utilizado o ajuste discreto, o controlador é discretizado pelo método de Tustin.

3.1.2 Ferramenta de Identificação

A ferramenta de identificação de processos é constituída por três métodos: Método gráfico, Método dos Mínimos Quadrados e Método dos Mínimos Quadrados Recursivo. Para o uso de qualquer um dos métodos, é necessário apenas conhecimento básico, no que diz respeito às técnicas de identificação. O usuário tem também a opção de importar dados de um processo real. Como a ferramenta não opera em tempo real, o Método dos Mínimos Quadrados Recursivo só pode ser utilizado com processos simulados.

A ferramenta permite:

- capturar dados de sistemas reais ou de simulações de outros ambientes;
- filtrar os sinais antes da identificação;
- simular ruídos no caso de plantas simuladas;

- realizar os experimentos de identificação e validação;
- comparar resultados;
- exportar os modelos obtidos para as etapas de projeto de controladores.

Método Gráfico

O método gráfico utiliza a resposta ao degrau do sistema na vizinhança do ponto de operação. Este método é bastante usado na indústria para o ajuste de controladores PID. Os dados podem ser importados de um ensaio real ou de uma simulação de outro ambiente. Se a opção é gerar dados na ferramenta, o usuário cria um processo simulado, aplica um degrau unitário na entrada e visualiza o gráfico da resposta no tempo do processo. A ferramenta indica quais os pontos que o usuário deve buscar no gráfico para obter o modelo desejado do processo (nesta versão, o modelo é restrito a 1ª ou 2ª ordem com atraso). A função de transferência do modelo identificado e seu erro quadrático são mostrados na tela, assim como a plotagem da resposta ao degrau do modelo, que pode ser comparada com a do processo. Na figura 3.2, é mostrada a janela correspondente a este método e em seguida parte dos seus códigos desenvolvidos em Matlab.

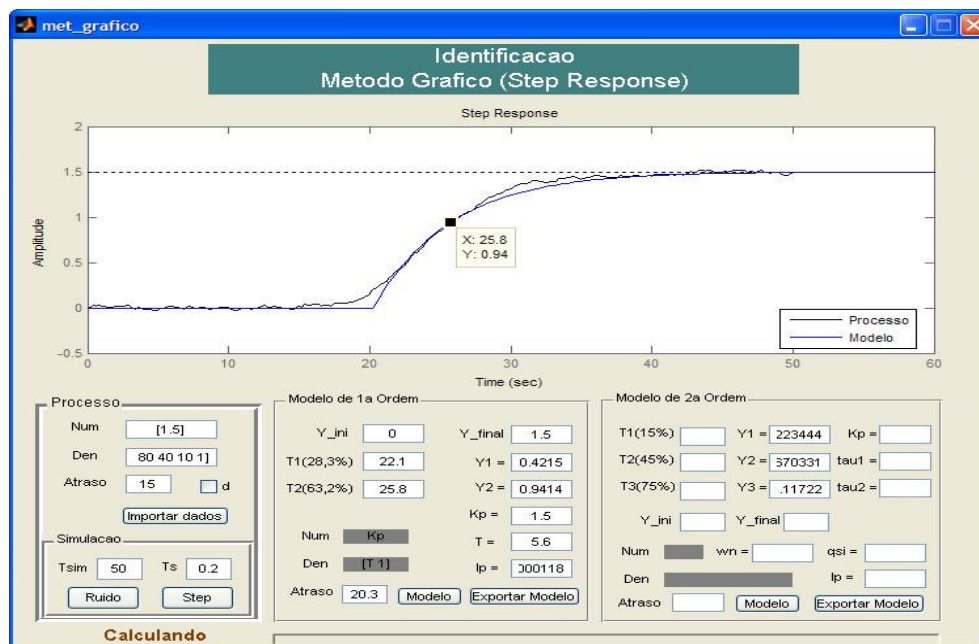


Figura 3.2: Método gráfico

Código do MATLAB®

```
% Pontos do processo
npt=size(yout);fim=npt(1);

ini=fix(fim*0.8);yest=yout(ini:fim);
npty=length(yest);soma=0;

for i=1:npty
    if i==1
        soma=yest(i);
    else
        soma=soma+yest(i);
    end
end

limite=soma/npty;

y1 = 0.283*limite;
y2 = 0.632*limite;

y11 = 0.15*limite;
y22 = 0.45*limite;
y33 = 0.75*limite;
```

Código do MATLAB®

```
% Calculo do Modelo de 1a ordem
Kp = str2num(get(handles.yfinal,'string'));
t1 = str2num(get(handles.t1,'string'));
t2 = str2num(get(handles.t2,'string'));

T = 1.5*(t2-t1);Ln = 1.5*(t1-1/3*t2);Pn = tf(Kp,[T 1],'inputdelay',Ln);

% Modelo de 2a ordem
Kp2 = str2num(get(handles.yfinal2,'string'));
t11 = str2num(get(handles.t11,'string'));
t22 = str2num(get(handles.t22,'string'));
t33 = str2num(get(handles.t33,'string'));

x=(t22-t11)/(t33-t11);qsi=(0.0805-(5.547*(0.475-x)^2))/(x-0.356);

if qsi < 1
f2=(0.708)*(2.811)^qsi;
else
f2=(2.6*qsi)-0.6;
end

wn=f2/(t33-t11);f3=(0.922)*(1.66^qsi);Ln2=t22-(f3/wn);

if qsi < 1
tau1=sym('tau1');tau2=sym('tau2');
eq1=tau1*tau2-1/wn^2;eq2=tau1+tau2-2*qsi/wn;
[tau1,tau2]=solve(eq1,eq2);
tau1=eval(tau1(1));tau2=eval(tau2(1));
else
tau1=(qsi+sqrt(qsi^2-1))/wn;tau2=(qsi-sqrt(qsi^2-1))/wn;
end

opttest=simset('SrcWorkspace','current','DstWorkspace','current');
sim('indicegraf',[0 tsim],opttest);

npts=tsimip/Ts;
Ip=(1/npts)*Ipgraf(end);
Ip=num2str(Ip);
```

Método dos Mínimos Quadrados

Os dados podem ser importados de um ensaio real ou simulados em outro ambiente. Para gerar dados na ferramenta, um sinal do tipo PRBS (*Pseudo-random Binary Signal*) é aplicado na entrada do processo simulado para a obtenção de dados. Para a identificação, o usuário deve introduzir uma estimativa do atraso do processo. Normalmente, este atraso estimado pode ser obtido de um teste de identificação gráfica num passo prévio. A partir destes dados, a ferramenta cria o modelo identificado, tanto na sua representação contínua como discreta. Nesta versão, o modelo é limitado à 2ª ordem com atraso dado que, na prática, estes modelos representam bem um conjunto muito grande de processos industriais monovariáveis. Após obtido o modelo, o usuário pode fazer a validação do mesmo. O método dos mínimos quadrados recursivo funciona de maneira similar, fornecendo um modelo de 1ª ordem com atraso.

3.1.3 Ferramenta de Controle

A ferramenta de controle possibilita trabalhar com três grupos de controladores:

1. PID: acadêmico, série e paralelo.
2. Compensadores de atraso (DTCs): Preditor de Smith, Observador de Perturbações e Preditor de Smith Filtrado.
3. Controladores Preditivos (MPCs): o GPC e o DTC-GPC.

As principais características são:

- todos os algoritmos de controle apresentam a mesma janela de interface;
- diferentes gráficos são plotados simultaneamente. Para cada controlador, o usuário dispõe de gráficos de resposta no tempo e de gráficos no domínio da frequência para análise de robustez. Isto permite otimizar os ajustes para cumprir compromissos de robustez e desempenho;
- em muitas situações da prática industrial, os controladores são apresentados para o operador em tempo contínuo, mas implementados em tempo discreto. A ferramenta contém esta opção, permitindo este tipo de análise e projeto;

- Para o ajuste de controladores, existem duas possibilidades: (i) o usuário avançado tem total liberdade para escolha de parâmetros e ordem dos controladores e modelos; (ii) o usuário iniciante pode usar os modelos e regras de ajustes sugeridas pelo programa;
- os dados podem ser salvos para uso em estudos comparativos (ver figura 3.3). Cada controlador dispõe de uma memória onde o usuário pode salvar resultados e decidir aquele que lhe convém para uma posterior comparação entre os demais controladores;
- todos os algoritmos possuem estrutura de dois graus de liberdade;
- é possível configurar os sinais de ruído a serem aplicados no processo sob controle.

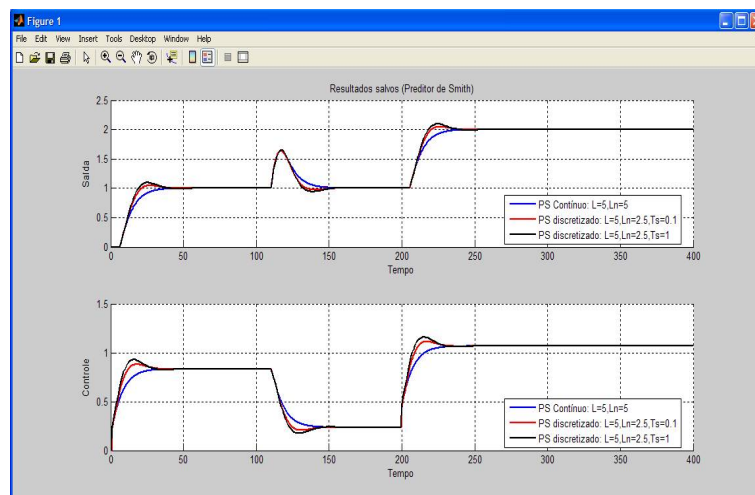


Figura 3.3: Comparação de resultados

A ferramenta ainda permite ao usuário escolher até três mudanças de referência no tempo de simulação e o tipo da perturbação (degrau, rampa, senóide ou ruído), tanto na entrada como na saída da planta, podendo esta passar por uma função de transferência. As interfaces destas configurações são mostradas na figura 3.4.

A perturbação padrão é aplicada na saída do processo, mas existe uma opção que, ao *clicar*, os dados do processo real são capturados automaticamente, fazendo com que a função de transferência da perturbação seja igual à do processo. O botão correspondente à essa opção é mostrada na figura 3.5. De acordo com a teoria de sistemas lineares, isso é o mesmo que aplicar a perturbação diretamente na entrada do processo. Caso o usuário não utilize nenhuma destas opções, a simulação é feita para um sinal de degrau unitário na entrada de referência e sem perturbação. Na próxima seção será mostrado um exemplo de uso da ferramenta.

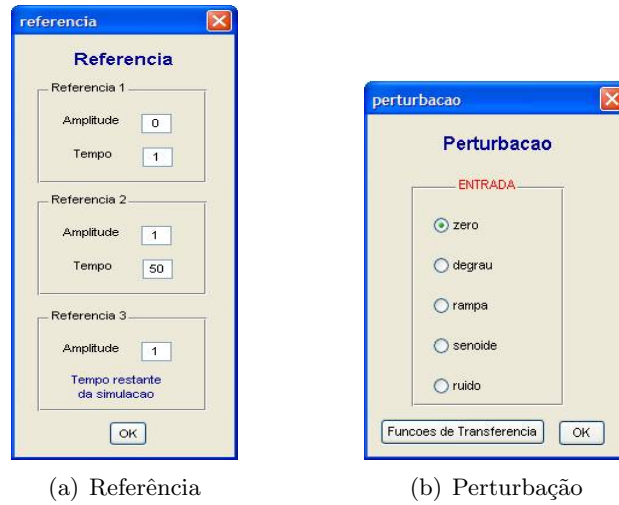


Figura 3.4: Interfaces de referência e perturbação

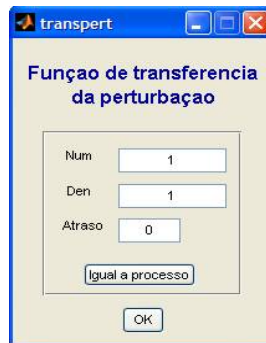


Figura 3.5: Função de transferência da perturbação

3.2 Aplicação

Nesta seção, será ilustrado o uso da ferramenta para um processo simulado, abordando dois métodos de identificação e duas estruturas de controle para serem comparadas. Parte da programação e do desenvolvimento da ferramenta também serão mostrados.

3.2.1 Identificação do processo

Considere o processo simulado por:

$$P(s) = \frac{1.5e^{-15s}}{(2s + 1)^5}$$

que representa a dinâmica de um sistema de vários processos de baixa ordem em cascata com um atraso de transporte. Supõe-se que o sinal de saída é afetado por um ruído branco de

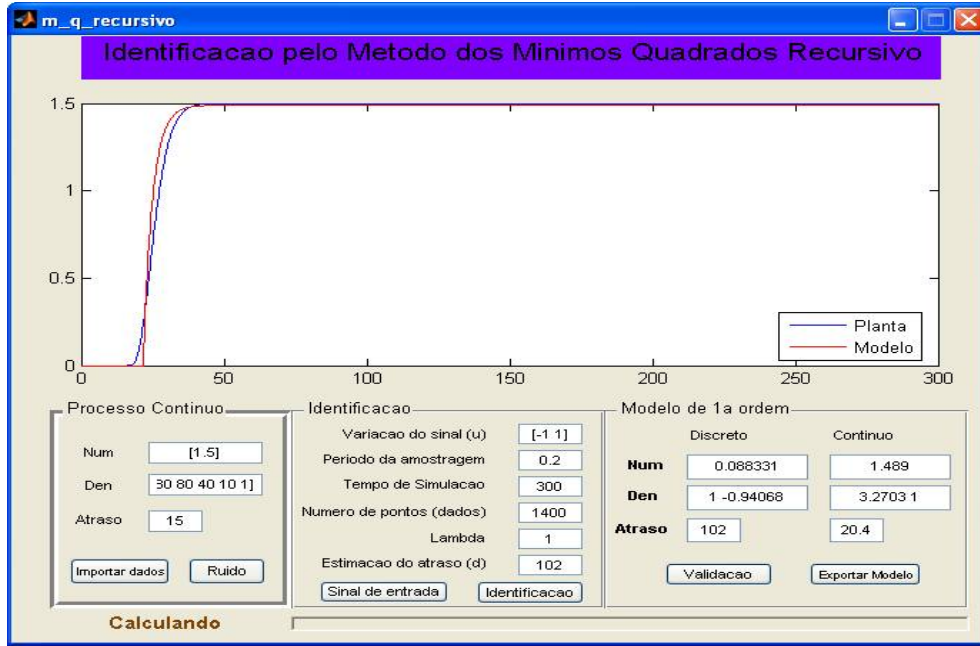


Figura 3.6: MMQ Recursivo

amplitude $A = 10^{-3}$. Assim, projeta-se um filtro para o ruído $F_{pb} = \frac{1}{1.5s+1}$ e escolhe-se um período de amostragem de $T_s = 0.2$ segundos. Utilizando o método gráfico para identificar um modelo a ser usado no controle, medem-se dois pontos da resposta ao degrau,

$$t_{28.3\%} = 22.1s \quad t_{63.2\%} = 25.8s$$

obtendo-se:

$$P_n = \frac{1.5e^{-20.3s}}{5.6s + 1}$$

e validando o modelo, um erro quadrático: $Ip = 1.18 \times 10^{-4}$.

Estes dados podem ser usados agora no método MQR, informando o atraso aproximado de 20 segundos já calculado. Na figura 3.6 é mostrado o resultado da simulação do modelo identificado e do processo. A validação do modelo obtido,

$$P_n = \frac{1.49e^{-20.4s}}{3.27s + 1}$$

é mostrada na figura 3.7. O erro quadrático desta validação é $Ip = 1.16 \times 10^{-4}$. Este modelo será usado nas simulações dos controladores.

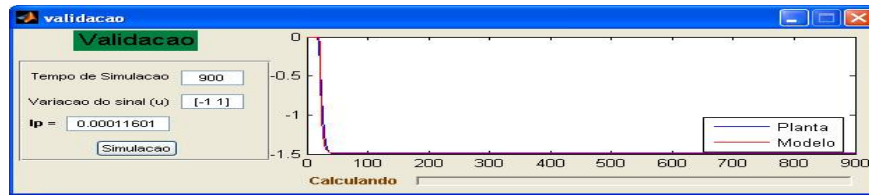


Figura 3.7: Validação

Código do MATLAB®

```
% Modelo de primeira ordem
d=get(handles.dest,'string');%Estimação do atraso
ga=str2num(get(handles.lambda,'string'));

u=u';am=zeros(1,npts);
bm1=zeros(1,npts);er1=0;
er=0;th=[am(1);bm1(1)];
I=eye(2);P=I;
d=str2num(d);

for k=d+2:npts

y(k)=yout(k);fi=[y(k-1);u(k-1-d)];
l=(P/ga)*fi/(1+fi'*P*fi/ga);
th=th+l*(y(k)-fi'*th);P=(I-l*fi')*P;
P=P/ga;
am(k)=th(1);bm1(k)=th(2);bm=bm1;
end;

ym=zeros(1,npts);n=npts;
for k=d+2:npts
ym(k)=am(n)*ym(k-1)+bm1(n)*u(k-1-d);
end;

b0 = bm1(end);a1 = am(end);
```

3.2.2 Controlador PID

O módulo do controlador PID permite três tipos de estrutura: acadêmica, série, paralela. Além dos ajustes tradicionais das ações proporcional, integral e derivativa, o PID inclui um filtro de referência e um filtro na sua parte derivativa, tornando-a realizável.

Diversos métodos de ajuste para o PID, considerando o atraso do processo, são encontrados na literatura. Entretanto, há alguns casos em que estes métodos não são válidos pelo fato de haver um atraso dominante do sistema.

Na ferramenta, existem quatro opções de ajuste para o PID: Ziegler-Nichols [13], Cohen-Coon [12], S-IMC (*simple-Internal Model Control*) [34] e o método do PS equivalente, visto no capítulo 2. O usuário tem também a opção de analisar o controlador no tempo discreto, o qual é implementado internamente pela ferramenta utilizando o método de Tustin; ou colocando diretamente os parâmetros discretos do controlador.

Caso seja escolhido um ajuste não apropriado ou não recomendado para o modelo, o usuário é alertado por uma janela de “erro” ou “atenção”. Neste caso, utilizou-se o método S-IMC, obtendo-se:

PID (Série):

$$K_c = 0.05, \quad T_I = 3.27, \quad T_D = 0 \quad \alpha = 0.05$$

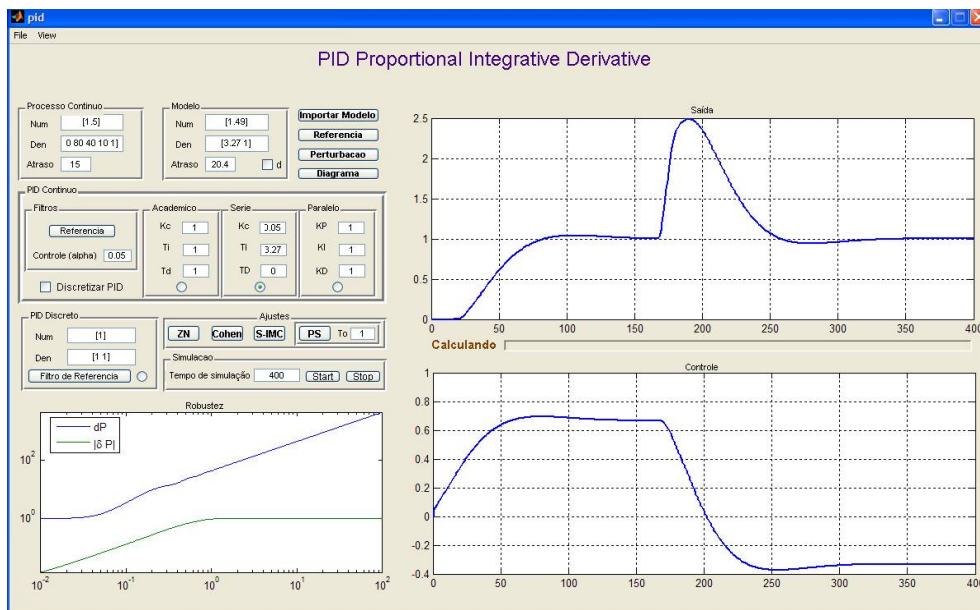


Figura 3.8: Simulação PID

Para a simulação, considera-se um degrau na referência em $t = 0$ e uma perturbação do tipo degrau unitário na entrada do processo em $t=150s$. Três gráficos são plotados: da saída, do controle e o de robustez como são mostrados na figura 3.8. No de robustez, o usuário visualiza o erro de modelagem e o índice de robustez, permitindo um melhor ajuste do controlador.

Código do MATLAB®

```
% Analise de robustez
% Caso Contínuo
ww=logspace(-2,2,500);
for ii=1:500
w=ww(ii);s=j*w;
Gn=polyval(numpn,s)/polyval(denpn,s);
Pn=Gn*exp(-Ln*s);
C=polyval(numc,s)/polyval(denc,s);
% robustness index
dP(ii)=abs((1+C*Pn)/(C*Pn));
% modeling error
Gr=polyval(nump,s)/polyval(denp,s);
Pr=Gr*exp(-L*s);
deltaP(ii)=abs(Pr/Pn-1);
end
loglog(ww,dP,ww,deltaP,'LineWidth',1)
axis([10^(-2) 10^(2) minP maxP]);
```

Código do MATLAB®

```
% Analise de robustez
% Caso Discreto
ww=logspace(-2,log10(pi/Ts),500);
for ii=1:500
w=ww(ii);z=exp(j*w*Ts);
Gn=polyval(numrnd,z)/polyval(denrnd,z);
Pn=Gn*z^(-dn);
C=polyval(numc,z)/polyval(denc,z);
% robustness index
dP(ii)=abs((1+C*Pn)/(C*Gn));
% modeling error
Gr=polyval(nump,j*w)/polyval(denp,j*w);
Pr=Gr*exp(-j*w*L);Pr=Pr*(1-1/z)/(j*w*Ts);
deltaP(ii)=abs(Pr/Pn-1);
end
loglog(ww,dP,ww,deltaP,'LineWidth',1)
axis([10^(-2) pi/Ts minP maxP]);
```

3.2.3 Controlador DTC-GPC

O controlador é projetado a fim de minimizar uma função custo sobre o horizonte de predição. O usuário tem como ajuste os seguintes parâmetros do controlador: horizonte de predição, horizonte de controle, ponderação do controle (λ) e o parâmetro β do filtro de robustez.

Como explicado no capítulo 2, a idéia do DTC-GPC é usar a estrutura de um DTC para o cálculo das predições e a do GPC para o cálculo do controle ótimo. A estrutura do DTC possui um filtro $R(z)$ no erro de predição. O ajuste deste filtro através do parâmetro β permite melhorar o índice de robustez do sistema [26].

Todas as condições de implementação do filtro $R(z)$ apresentadas na seção 2.5.2 são calculadas internamente pela ferramenta, como sua ordem e parâmetros associados ao β , garantindo a estabilidade interna do sistema. Desta forma, o usuário apenas ocupa-se do ajuste através da visualização gráfica do erro de modelagem e do índice de robustez.

Para o processo analisado, utilizam-se os seguintes dados de entrada, para um periodo de amostragem $T_s = 0.5$:

Horizonte de predição: $N = 10$

Horizonte de controle: $N_u = 10$

$\lambda = 2$

$\beta = 0.75$

O filtro é, neste caso:

$$R(z) = \frac{(1 - \beta)^2}{(z - \beta)^2}$$

E usam-se os mesmos sinais de entrada que no caso do PID.

Após a análise dos resultados apresentados na figura 3.9, o usuário pode, se desejar, comparar as simulações numa nova janela onde são calculados os respectivos IAE e ISE (ver figura 3.10). Como era esperado, o DTC-GPC apresenta melhor resultado, já que o controlador compensa o atraso da planta.

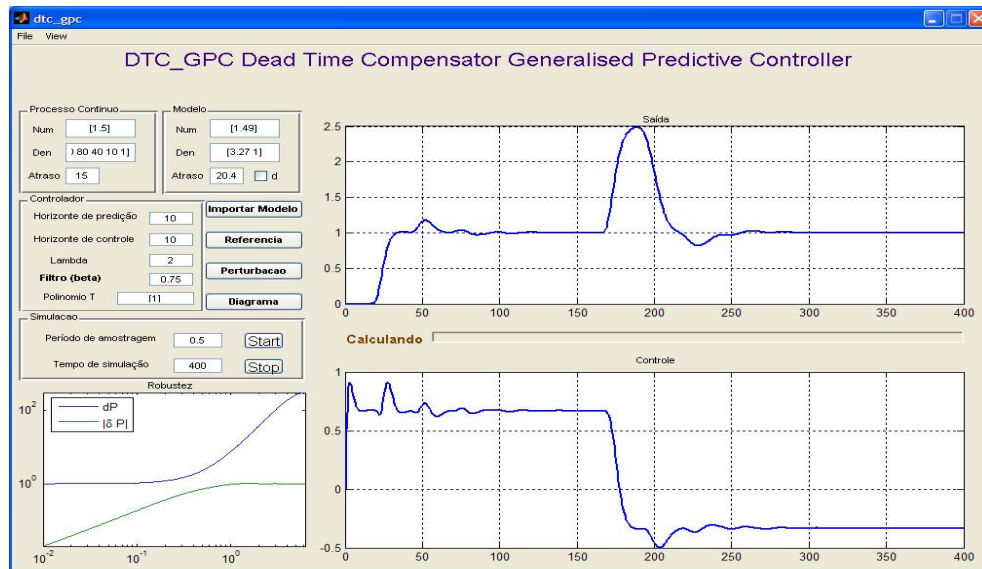


Figura 3.9: Simulação do DTC-GPC

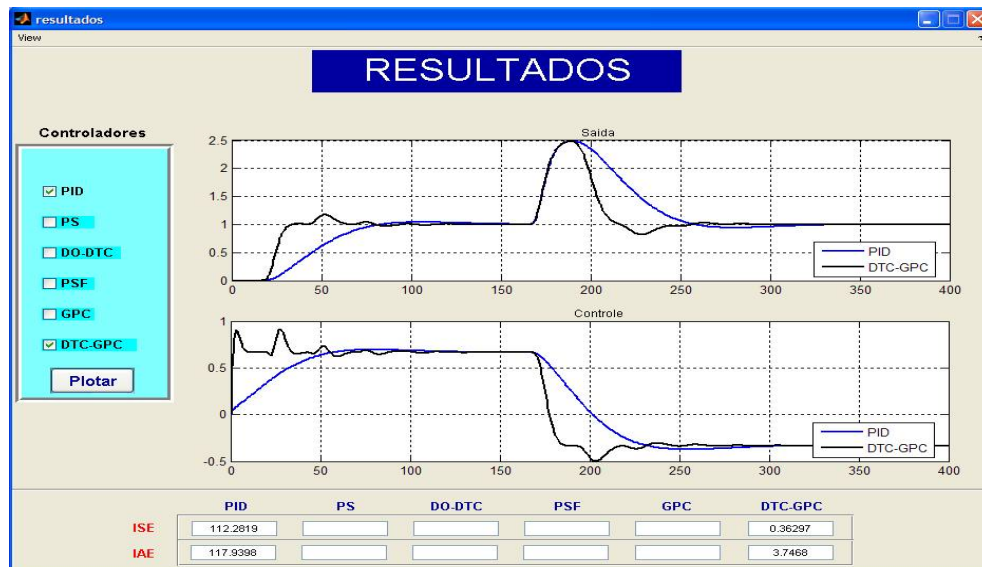


Figura 3.10: Resultados

```

% Filtro de Robustez
beta = get(handles.beta,'string');
switch beta
case 'Sem Fr'
Nf=1;Df=1;
otherwise
beta=str2num(beta);polos(1,:)=roots(A);
m11=length(A)-1;

for i=1:m11
if abs(polos(i))>=1 %verifica se tem pólos instáveis
run('filtro-fr') %roda programa p/ calcular o filtro
break;
elseif m3==1 %p/ perturbação do tipo senóide
run('filtro-fr2') %roda programa p/ calcular o filtro
else
Nf=(1-beta)^2;Df=[1 -2*beta beta^2];
end
end
% implementação do bloco S
Dfz=[Df zeros(1,dn)];
dd=size(Dfz)-size(Nf);dd=dd(2);
numS1=Dfz-[zeros(1,dd) Nf];
aux1=numS1;aux2=A;
run('mdc') %roda programa que calcula MDC
[pol1,r1]=deconv(numS1,mcd);
[pol2,r2]=deconv(A,mcd);
numS=conv(B,pol1);denS=conv(Dfz,pol2);
end

```


Capítulo 4

A Ferramenta aplicada em projetos de controle em uma usina de produção de açúcar

A exemplo de outros setores industriais, a automação dos processos é evidente em uma usina de produção de açúcar. Através de sistemas digitais de controle, é possível distribuir-se geograficamente as facilidades oferecidas pelo computador a um custo aceitável, pois permitem que o controle global e integrado de uma unidade industrial seja feito em etapas.

A incorporação de um sistema de armazenamento permite avaliar as variações dos dados da usina ao longo do tempo, possibilitando um ajuste de controle que garanta um regime razoavelmente uniforme no setor. Neste capítulo, será mostrado um trabalho realizado em parceria com a usina *Ingenio La Unión, S.A.*, situada na cidade de Escuintla, na Guatemala. Com os dados fornecidos pela usina, a ferramenta será utilizada para modelar, projetar e simular os laços de controle de seus processos.

4.1 Introdução

A cristalização por resfriamento constitui um importante processo na produção de açúcar onde é produzida a última extração de sacarose do licor mãe e máximo esgotamento do mesmo. Como é conhecido nas fábricas de açúcar, a cristalização da sacarose ocorre, fundamentalmente, em reatores à vácuo. No entanto, quando se trata de cozimento de massas

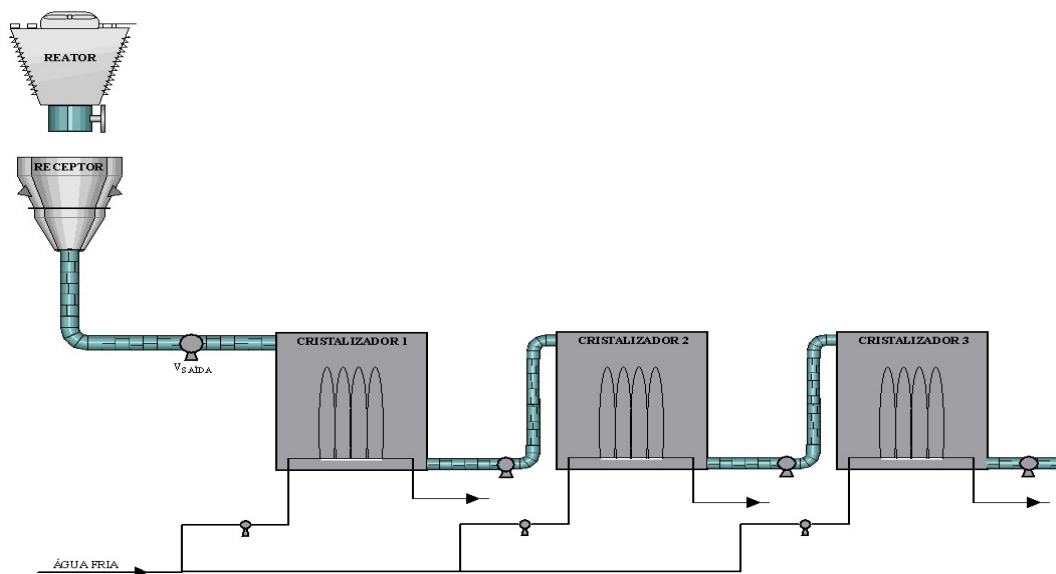


Figura 4.1: Processo de cristalização

de baixas purezas, nestes equipamentos, o esgotamento do licor mãe não pode ser levado até o ponto desejado. Isto é devido ao fato de que, em cozimentos de baixas purezas, a velocidade de cristalização sofre uma brusca diminuição, em consequência das altas viscosidades produzidas quando o licor mãe se aproxima do esgotamento. Por esta razão, torna-se necessário continuar o processo de cristalização através de outros métodos e equipamentos, com o objetivo de reduzir ao mínimo a quantidade de sacarose dissolvida no licor mãe.

Nos cristalizadores por esgotamento, prossegue a cristalização iniciada no reator (ver figura 4.1). A diferença é que neles a cristalização se realiza pela diminuição da solubilidade da sacarose, em função do esfriamento gradual no qual é submetida a massa cozida. A solubilidade da sacarose no licor mãe se reduzirá à medida que a temperatura diminui. Uma parte da sacarose dissolvida ficará em solução supersaturada e a mistura, em condições apropriadas, depositar-se-á sobre os cristais já existentes. Este tratamento assegura um maior esgotamento do licor mãe, melhorando o índice de recuperação da fábrica.

O cristizador é um trocador de calor que utiliza água fria para baixar a temperatura da massa nele contida. Basicamente, dois fatores importantes influenciam na sua dinâmica de troca de calor: a vazão de água fria na entrada e o nível de massa no qual se encontra.

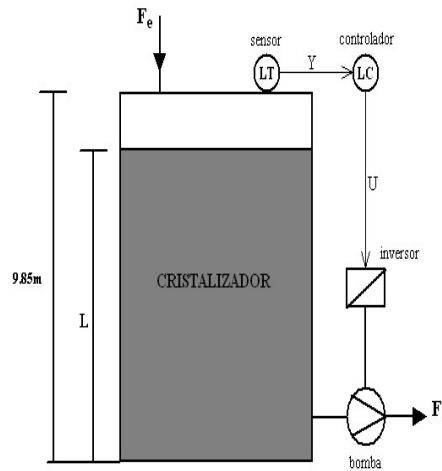


Figura 4.2: Cristalizador e componentes

4.2 Projeto do controle de nível

Em cada cristalizador é necessário implementar um controle de nível que garanta tanto o balanço de massa como a eficiência térmica do processo de esfriamento. O objetivo do controle é manter o nível L do cristalizador de acordo com uma referência desejada através da variável manipulada, o fluxo de saída da massa $F_s(s)$. As variações do fluxo de entrada da massa $F_e(s)$ constituem a perturbação principal cujo esquema do controle de nível é mostrado na figura 4.2. A malha de controle é constituída pelos seguintes componentes: sensor de nível (LT), controlador (LC), variador, motor e bomba.

O processo de cristalização por resfriamento em um conjunto de cristalizadores é um processo contínuo e como tal, seu rendimento ótimo se consegue com uma alimentação constante. Partindo desta premissa, a melhor solução seria implementar um controle de fluxo de saída do receptor, ou seja, medir o fluxo de massa entrando no cristalizador e atuar sobre a velocidade da bomba para garantir o fluxo desejado. No entanto, este tipo de controle é difícil, pois exige que as massas de terceira (massas no terceiro estágio de processamento) se produzam com uma periodicidade absolutamente constante ou que o receptor tenha uma grande capacidade de volume, o que não ocorre na maioria das vezes. O processo de produção nos tanques é por etapa, o que gera uma descontinuidade do processo, tornando também a periodicidade com que sucedem as massas de terceira variável. Por estas razões, uma alternativa do sistema de controle de alimentação consiste em implementar um laço de controle de nível da massa cozida no receptor de massas de terceira, projetado para produzir o máximo de amortecimento possível das oscilações do fluxo de alimentação e ao mesmo tempo garantindo que o receptor não transborde ou fique completamente vazio. Como os reatores à vácuo são

do tipo batelada, a descarga no receptor é cíclica e, portanto, pode ser aproximada por uma onda quadrada. Desta forma, é necessário levar em conta que este sistema não garante um fluxo constante. Apenas garante um fluxo contínuo que será oscilatório ou periódico, ou seja, o fluxo de massa de entrada do primeiro cristalizador será cíclico ou periódico. Assim, quanto maior a capacitância e o volume do receptor de massa, mais suaves serão as variações cíclicas do seu fluxo de entrada. Isto significa que tanto o controle de temperatura como o controle de nível de cada cristalizador estarão submetidos a uma perturbação cíclica que, supostamente, tem suas desvantagens.

4.2.1 Modelagem do nível do cristalizador

Considerando a equação de balanço de massa de cada cristalizador, tem-se:

$$\rho_e F_e(t) - \rho_s F_s(t) = \frac{dm}{dt} \quad (4.1)$$

$$m(t) = A \rho_p L(t) \quad (4.2)$$

onde F_e é o fluxo de entrada da massa, F_s é o fluxo de saída da massa, m é a massa dentro do cristalizador, A é a área transversal do cristalizador e L é o nível. As densidades de entrada, de dentro e de saída do cristalizador são, respectivamente, ρ_e , ρ_p e ρ_s .

Supondo $\rho_e = \rho_p = \rho_s$ e substituindo (4.2) em (4.1), obtém-se:

$$F_e(t) - F_s(t) = A \frac{dL}{dt} \quad (4.3)$$

Considerando as condições iniciais nulas e aplicando a transformada de Laplace, tem-se a seguinte função de transferência do nível:

$$L(s) = \frac{F_e(s)}{As} - \frac{F_s(s)}{As} \quad (4.4)$$

O diagrama de blocos que descreve a dinâmica do nível sujeito às possíveis variações de fluxos de entrada e saída é mostrado na figura 4.3.

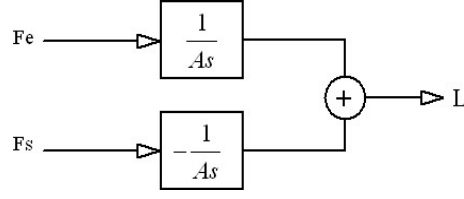


Figura 4.3: Modelo dinâmico do nível de cada cristalizador

4.2.2 Modelagem do conjunto bomba-variador de frequência

A dinâmica do fluxo bombeado em função do sinal de controle, aplicado sobre o variador, pode ser representada com exatidão por uma equação diferencial de primeira ordem:

$$\tau_a \frac{dF_s}{dt} + F_s(t) = k_a U(t) \quad (4.5)$$

onde $U(t)$ é o sinal de controle.

Considerando as condições iniciais nulas e aplicando a transformada de Laplace, tem-se:

$$F_s(s) = \frac{k_a}{\tau_a s + 1} U(s) \quad (4.6)$$

A constante de tempo τ_a e o ganho k_a devem ser determinados experimentalmente. Usando o fluxo máximo (faixa de operação do variador a 100%), obteve-se um ganho de $0.615 \text{ m}^3/\text{min}\%$ e de $0.518 \text{ m}^3/\text{min}\%$ nos atuadores de saída e entrada, respectivamente; e com a mesma constante de tempo, $\tau_a = 0.5 \text{ min}$.

4.2.3 Modelagem do sensor de nível

A dinâmica da saída do sensor em função das variações do nível pode ser representada por uma equação de primeira ordem:

$$\tau_m \frac{dy}{dt} + y(t) = k_m L(t) \quad (4.7)$$

Considerando as condições iniciais nulas e aplicando a transformada de Laplace, tem-se:

$$Y(s) = \frac{k_m}{\tau_m s + 1} L(s) \quad (4.8)$$

Da mesma forma como no caso anterior, a constante de tempo τ_m e o ganho k_m devem

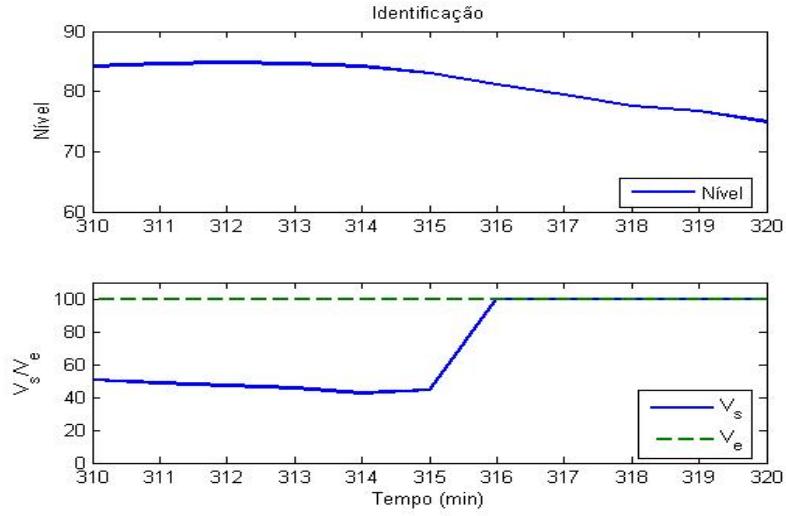


Figura 4.4: Ensaio de identificação da dinâmica do nível

ser determinados experimentalmente. Para os sensores de nível, obteve-se um ganho unitário e uma constante de tempo de 0.2 minutos.

Sendo assim, o modelo teórico final é:

$$L(s)[\%] = \frac{k}{s(\tau_a s + 1)(\tau_m s + 1)} [V_e - k_s V_s],$$

$$com \begin{cases} k_s = \frac{0.518}{0.615} \\ k = 1/A \end{cases}$$

onde V_e é o variador de entrada e V_s é o variador de saída.

As dinâmicas do atuador e do medidor podem ser desprezadas, dado que são muito mais rápidas que a dinâmica do nível. Logo, o modelo usado será:

$$L(s)[\%] = \frac{K}{s} [V_e - k_s V_s]$$

Como o resultado teórico está sujeito à erros de medição, foi realizado um ensaio experimental de identificação para o ajuste da constante K , cuja dinâmica é mostrada na figura 4.4.

Com os dois variadores em 100% durante vários minutos, mediu-se a variação do nível.

Desta forma, o ganho do sistema foi calculado como:

$$K = -1.06 \cdot 10^{-3} [1/min],$$

obtendo-se o seguinte modelo:

$$P_{n1}(s)[\%] = \frac{L(s)[\%]}{F_s(s)[\%]} = \frac{-1.06 \cdot 10^{-3}}{s}$$

4.2.4 Simulação

A estrutura do laço de controle é mostrada na figura 4.5. Observa-se que a ação do controle é inversa devido ao fato do ganho do processo ser negativo.

Como visto no capítulo 3, a estrutura de controle implementada na ferramenta é diferente da estrutura da usina. Porém, é possível, através de algumas manipulações, tornar a estrutura da usina equivalente à da ferramenta como ilustrada na figura 4.6, onde $G_n(s)$ é a função de transferência do sensor e o controlador tem ganho negativo.

Assim, as configurações do processo e perturbação utilizadas na ferramenta são, respectivamente:

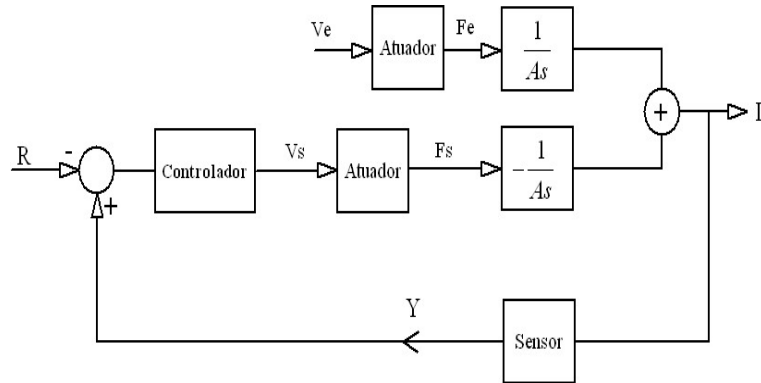


Figura 4.5: Diagrama de blocos do controle de nível de cada cristalizador

$$P(s) = \frac{-1}{As} \frac{k_a}{(\tau_a s + 1)} \frac{k_m}{(\tau_m s + 1)} \quad (4.9a)$$

$$Q(s) = \frac{1}{As} \frac{k'_a}{(\tau_a s + 1)} \frac{k_m}{(\tau_m s + 1)} \quad (4.9b)$$

Substituindo os valores das dinâmicas dos medidores e atuadores, obtidos experimentalmente, e as dimensões dos cristalizadores ($A = 66m^2, L_{max} = 9.85m$) nas equações (4.9a)

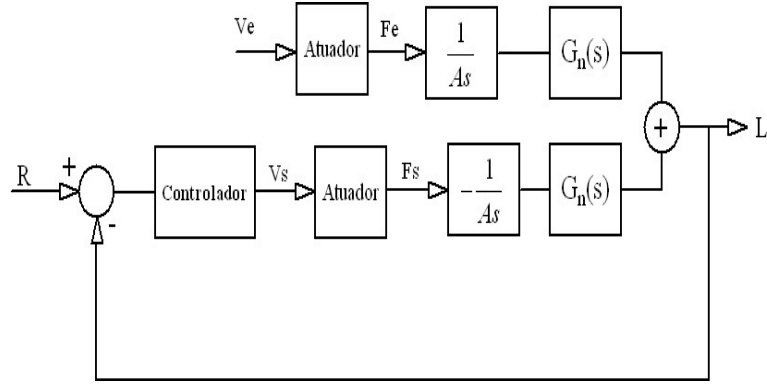


Figura 4.6: Diagrama de blocos equivalente do controle de nível

e (4.9b), a relação de entrada e saída, em %, é dada por:

$$P(s)[\%] = \frac{-9.46 \times 10^{-4}}{s(0.5s + 1)(0.2s + 1)} \quad (4.10)$$

e a função de transferência da perturbação é

$$Q(s)[\%] = \frac{7.97 \times 10^{-4}}{s(0.5s + 1)(0.2s + 1)} \quad (4.11)$$

Ajuste do controlador

Considerando um modelo integrativo, $P_{n2}(s)[\%] = \frac{-9.46 \cdot 10^{-4}}{s}$, que pode ser obtido a partir de (4.10), o controlador de nível é ajustado, impondo uma dinâmica de malha fechada com um tempo de resposta de 20 minutos e sem oscilações. Para isto, é utilizado um controlador PI com um filtro de referência.

O controlador *PI Foxboro*, utilizado pela usina, tem a seguinte lei de controle:

$$u = \frac{100}{P} \left[\left(\frac{1}{I_s} + A_c \right) r - \left(\frac{1}{I_s} + 1 \right) c_f \right] f_r + b \quad (4.12)$$

onde P é a banda proporcional, I é o tempo integrativo em minutos, r é a referência, A_c é a ponderação da referência, c_f é a medição filtrada, f_r é um filtro passa-baixa e b é o *bias*.

Considerando $b = 0$ e $f_r = 1$, esta estrutura equivale a um controlador PI tradicional

$G_c(s)$ com um filtro de referência, $G_f(s)$, dados por:

$$G_c(s) = \frac{100}{P} \frac{Is+1}{Is}$$

$$G_f(s) = \frac{A_c Is+1}{Is+1}$$

Com este controle e o modelo identificado anteriormente, a função de transferência em malha fechada é dada por:

$$MF = \frac{A_c Is + 1}{\frac{PI}{100K} s^2 + Is + 1}, \text{ onde } K = 9.46 \cdot 10^{-4}$$

e sua equação característica para $T_o = 20$:

$$\Delta(s) = T_o^2 s^2 + 2T_o s + 1$$

$$\Delta(s) = 400s^2 + 40s + 1$$

Calculando-se os parâmetros em MF, obtém-se:

$$I = 40 \quad P = 0.95$$

Por fim, o valor de A_c é calculado para evitar o efeito de um dos pólos ($p_1 = p_2 = -0.05$) de malha fechada:

$$A_c Is + 1 = 0$$

$$A_c \cdot 40 \cdot (-0.05) + 1 = 0$$

$$A_c = 0.5$$

Para representar a perturbação cíclica na ferramenta, utilizou-se uma perturbação senoidal com 35% de amplitude e período de 40 minutos.

A simulação do comportamento do sistema para esse ajuste de controle é mostrada na figura 4.7. É importante salientar que, além dos valores de saída do processo e do sinal de controle serem dados em valores percentuais, os mesmos também são incrementais, já que é considerado que o sistema esteja atuando próximo ao ponto de operação. Na tentativa de aproximar ainda mais da situação prática, foi assumido que o modelo P_{n1} correspondia ao processo real operado, de fato, na usina. Observa-se que é obtida uma resposta bastante satisfatória com poucos desvios em relação ao *setpoint* desejado.

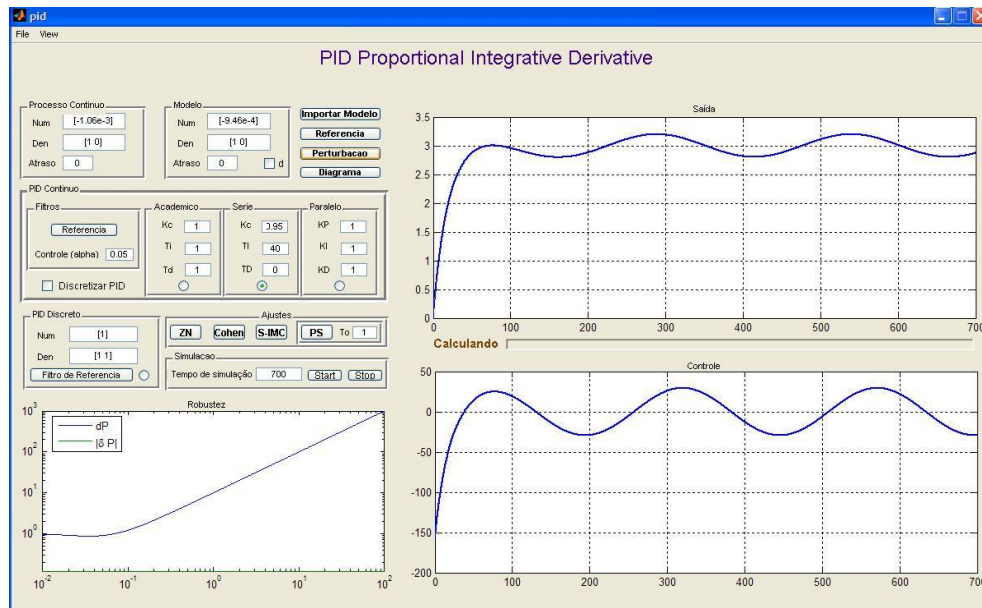


Figura 4.7: Simulação do controle de nível

4.2.5 Experimentos realizados

Para ilustrar os benefícios do novo ajuste do controle de nível, serão mostrados, em seguida, alguns experimentos realizados na usina.

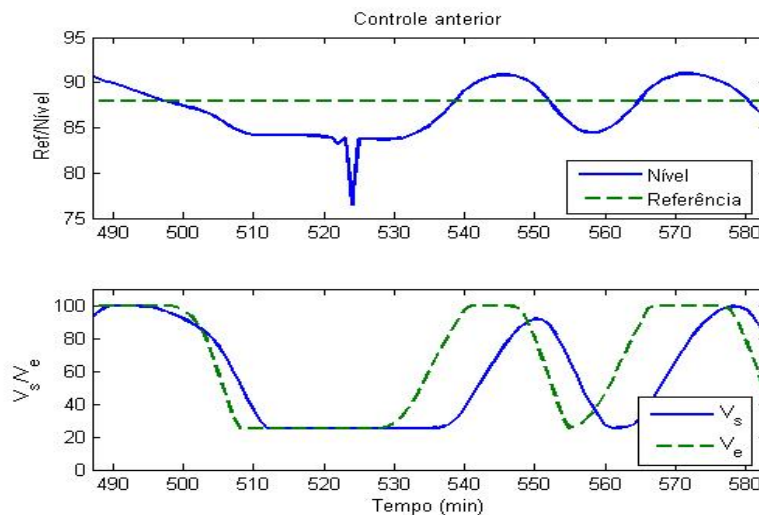


Figura 4.8: Controle de nível anterior

Experimento 1: Controle de nível antes do ajuste do controlador. Com o PI que estava sendo utilizado na usina, o sistema oscilava bastante em torno do *setpoint* de 88% de acordo com os resultados da figura 4.8. Conforme constatado, existe uma proteção do sistema que

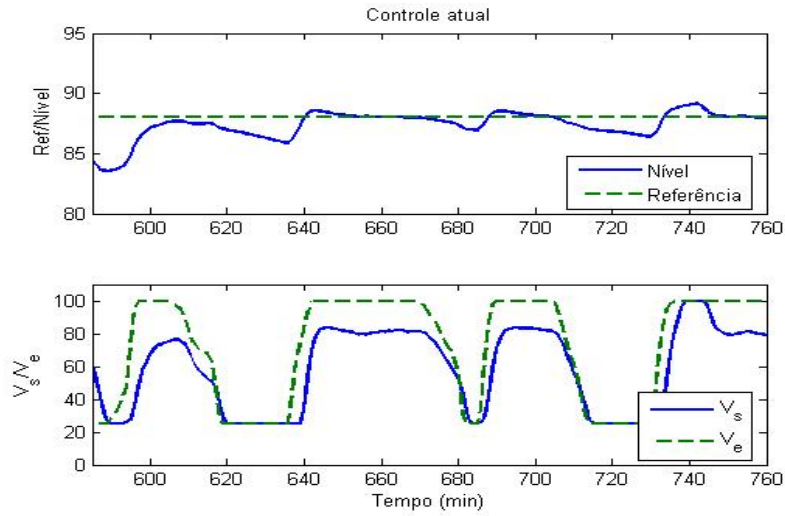


Figura 4.9: Controle de nível atual

impede que a vazão seja inferior a 25%. Isto evita um interrompimento das bombas devido à viscosidade da massa.

Experimento 2: Controle de nível com o novo ajuste do controlador. Os resultados da figura 4.9 comprovam o que havia sido simulado pela ferramenta. O controlador mantém o nível em torno do *setpoint* de 88% com uma oscilação muito menor. Observa-se que a vazão de entrada é quase uma onda quadrada, perturbando o sistema de forma cíclica.

Experimento 3: Controle de nível com o novo ajuste do controlador. Na figura 4.10, é mostrado como o sistema rejeita as perturbações geradas pelas oscilações de entrada de massa.

4.3 Projeto do controle de temperatura

O controle de temperatura implementado consiste em medir a temperatura de massa cozida na saída de cada cristalizador e atuar sobre o fluxo de água na entrada. O sistema de esfriamento opera como um trocador de calor.

O processo de esgotamento do mel, do ponto de vista de controle, contém as seguintes perturbações que afetam a temperatura da massa cozida: (a) as trocas do fluxo de massa cozida fornecida na entrada do cristalizador; (b) as variações de temperatura de entrada da massa nos tanques; (c) da temperatura ambiente; (d) da temperatura da água de resfriamento.

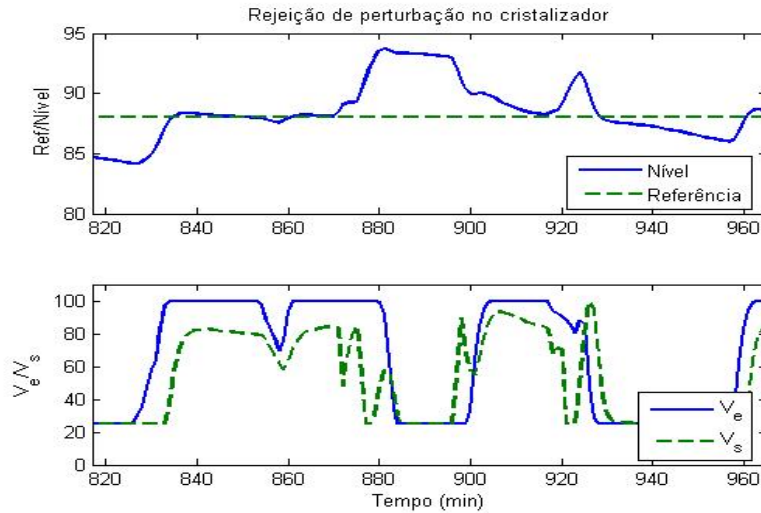


Figura 4.10: Rejeição de perturbações no cristalizador

As variações da temperatura ambiente e da temperatura da água de resfriamento têm um efeito desprezível no sistema, salvo em períodos frios do ano.

A dinâmica da temperatura de massa cozida, decorrente das mudanças de fluxo de água, é extremamente lenta devido às altas capacidades de massa e energia que os cristalizadores verticais possuem. Para otimizar o processo de cristalização por resfriamento, é necessário garantir uma supersaturação constante durante todo o processo de resfriamento, aumentando o máximo possível sua velocidade e evitando, assim, a formação de falsos grãos de açúcar. Para isto, deve-se garantir uma velocidade de resfriamento constante da massa cozida nos cristalizadores. A primeira solução é avaliar e projetar o sistema de controle atual implementado pela usina. O critério do projeto é reduzir as variações da temperatura em torno do seu valor ótimo, ajustado pelo controlador.

4.3.1 Modelagem e identificação do sistema

O processo é basicamente um trocador de calor que, com atuação na vazão de água fria na entrada, controla-se a temperatura da massa na saída.

Para a modelagem do sistema de resfriamento, foi aplicado um degrau, variando de 50% para 100% o fluxo de água na entrada do cristalizador, onde observou-se o comportamento de sua dinâmica durante um período de 2000 minutos. A entrada (em *gal/min*) e resposta (em $^{\circ}F$) do sistema estão representadas na figura 4.11.

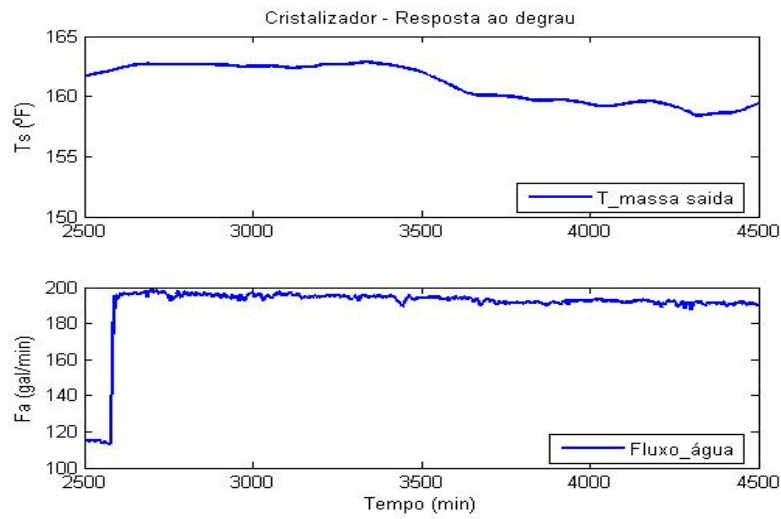


Figura 4.11: Resposta ao degrau

Importando os dados e ajustando os parâmetros do processo na ferramenta, de acordo com as figuras 4.12 e 4.13, foi obtido o seguinte modelo:

$$G_t(s) = \frac{-1.883 \times 10^{-2}}{240s + 1} e^{-925s}$$

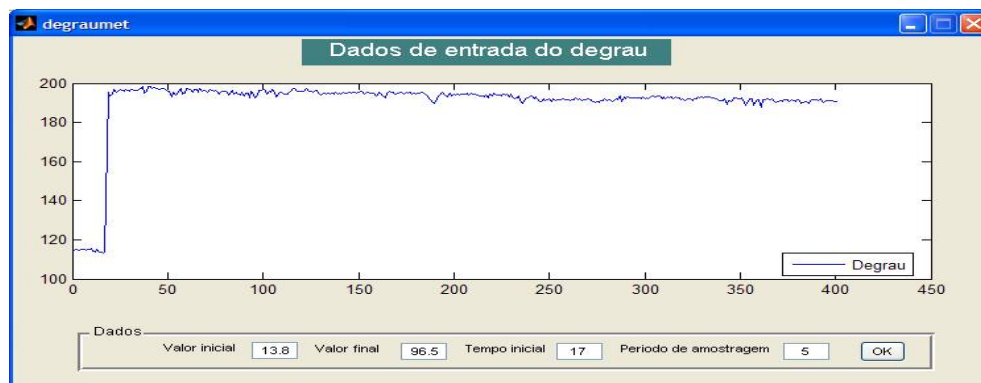


Figura 4.12: Dados de entrada

Estes dados mostram que o sistema tem um ganho muito pequeno, o que indica que sua capacidade de refrigeração compromete um desempenho satisfatório do controle. Todavia, serão feitas algumas simulações nas quais servirão como base para um projeto futuro do controle, já que no momento, pelo fato da usina ainda estar construindo um melhor sistema de resfriamento, o controle é operado em modo manual.

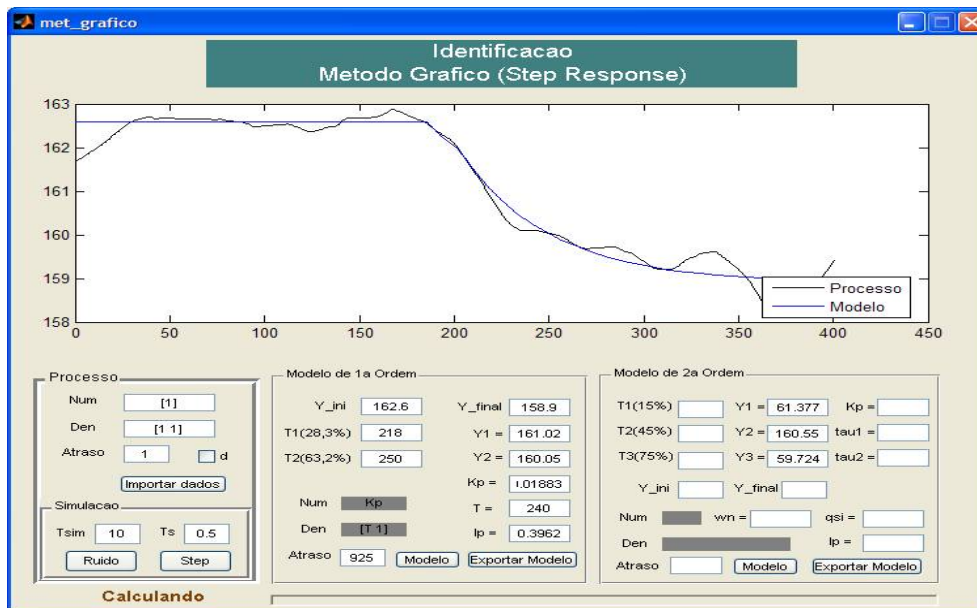


Figura 4.13: Identificação da dinâmica da temperatura

4.3.2 Simulação

Como a ferramenta permite diversos tipos de ajuste automático e estrutura, tendo o PID como controlador, foram feitas apenas duas comparações de resultados obtidos. Na figura 4.14, é mostrado o comportamento do sistema para um *PID-PS* com um $T_0 = 20$ e na figura 4.15, compara-se com a resposta cujo controle é ajustado com o *PID S-IMC*. Em ambos os casos, o *setpoint* é configurado para que a temperatura caia $3^\circ F$. Considera-se uma perturbação do tipo degrau em $t = 4500 \text{ min}$ e que o sistema esteja atuando no ponto de operação $T = 160^\circ F$. Portanto, os dados de saída ($^\circ F$) e controle (gal/min) são valores incrementais.

Nota-se que o PID tem ganho negativo. Assim, deve ser implementado no *PID Foxboro* com modo inverso. Os resultados mostram que o *PID S-IMC* obtém uma resposta sem oscilações, porém muito lenta. Já o *PID-PS* apresenta uma resposta bem mais rápida com algumas oscilações. Um reajuste do *PID-PS*, na prática, pode levar a um melhor compromisso tempo-oscilações.

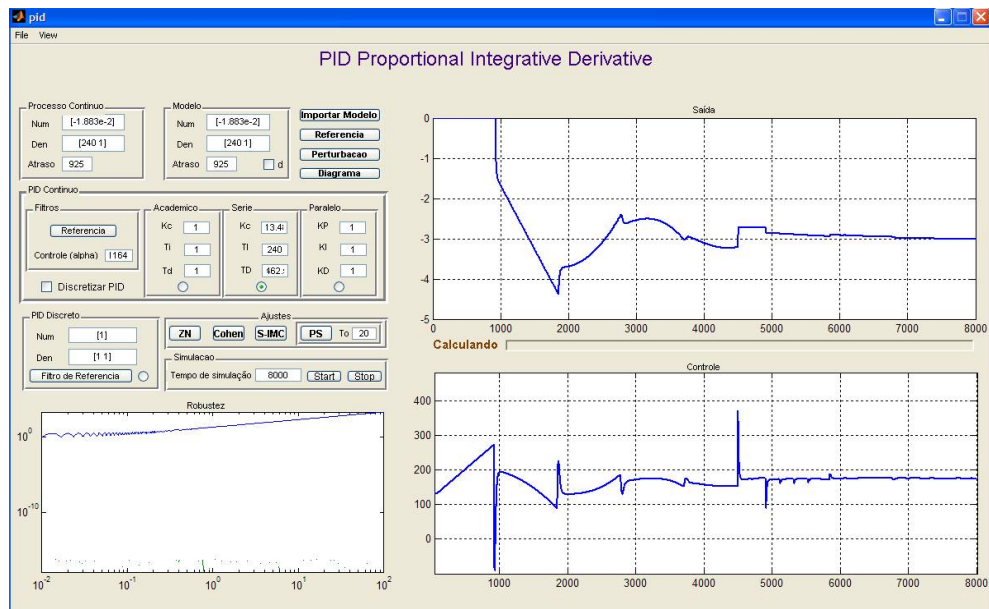


Figura 4.14: Ajuste PS

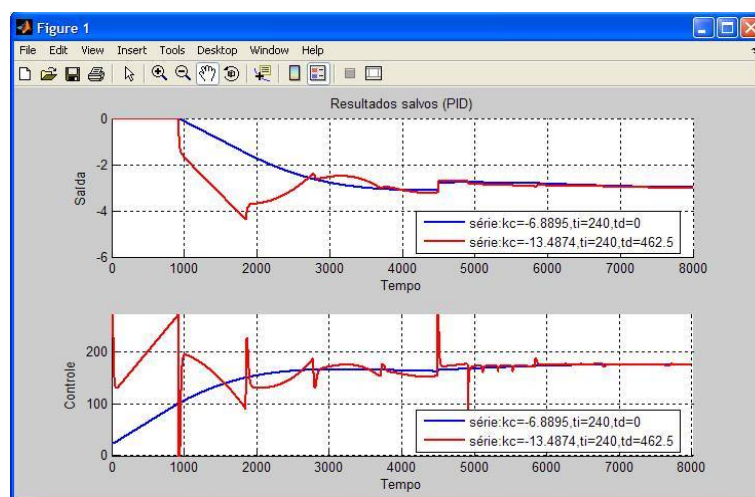


Figura 4.15: Comparação

Capítulo 5

Conclusões

A evolução do ambiente acadêmico torna sua relação com a área tecnológica ainda mais estreita, o que implica em vocábulos como produtividade, qualidade e custo, cada vez mais presentes no cotidiano dos profissionais envolvidos nesse meio. Pensando nisso, a ferramenta foi desenvolvida, objetivando uma junção entre os aspectos teóricos relacionados aos sistemas com atraso de transporte e a realidade industrial, garantindo, consequentemente, um melhor rendimento para o engenheiro ou estudante.

Foram feitos diversos testes de funcionalidade e aplicações, tanto na área acadêmica quanto na área industrial, com uma aceitação bastante satisfatória por parte dos usuários, em decorrência dos resultados obtidos. Todavia, nenhum protótipo de software está imune a erros e, portanto, este projeto não está isento de tal fator, o que se torna ainda mais justificável frente ao número de algoritmos envolvidos no trabalho.

Do ponto de vista de controle, neste documento, o usuário tem acesso aos principais tópicos de assuntos relacionados aos processos com atraso. Foram revisados alguns ajustes do PID, do DTC, assim como abordados algumas estratégias de controle preditivo, mostrando suas vantagens e desvantagens. A fim de ajudar o usuário a se familiarizar com a ferramenta, foram ilustrados diversos exemplos com os códigos do Matlab expostos.

Destaca-se a importância da simulação como auxílio na análise e estudo de novas técnicas de controle aplicadas. Há um desenvolvimento mútuo entre as ferramentas computacionais e as novas metodologias aplicadas no que diz respeito às teorias de controle, em função da relação simbiótica entre programas e usuários, estes atraídos a descobrir os recursos implementados.

Atualmente, este trabalho está publicado no site do livro *Control of Dead-time Processes* e um artigo foi aceito no “XVII Congresso Brasileiro de Automática (CBA 2008)” . Porém, a tendência é ainda publicá-lo em algumas revistas destinadas à comunidade de controle e no site da *Mathworks*, fabricante do Matlab.

Concluir este trabalho significa marcar um ponto de continuidade no desenvolvimento de ferramentas de auxílio ao engenheiro de controle, permitindo uma integração maior entre as soluções computacionais e a área de Engenharia de Controle. O fato da ferramenta conter código aberto evita a perda de rotinas implementadas, facilitando suas reutilizações e possibilitando uma extensão que permita integrar algumas ferramentas já existentes, podendo gerar um produto ainda mais poderoso.

Sendo assim, fica, então, como perspectivas de trabalhos futuros os seguintes tópicos relacionados abaixo:

- diminuição de erros;
- otimização dos algoritmos;
- desenvolvimento de novas interfaces;
- extensão da ferramenta para os casos de sistemas multivariáveis e não-lineares;
- implementações de novos pacotes, contendo outros métodos de identificação e outras estruturas de controle;
- abordagens que envolvam análises da resposta no domínio da frequência.

É importante salientar que está associado ao trabalho um manual, produzido com o intuito de permitir a aplicação correta da ferramenta. Esta que tem uma interface agradável e de uso intuitivo, facilitando a simulação e comparação de controladores, dos mais simples aos mais avançados; e ainda permitindo ao estudante ou engenheiro o aprendizado e utilização adequada dos controladores.

Enfim, por integrar as principais estruturas de controle no meio industrial, assim como algoritmos de identificação, torna-se um produto único: **Uma Ferramenta de Apoio ao Projeto de Sistemas de Controle para Processos com Atraso.**

Apêndice A

Manual

Além de ser necessário um conhecimento básico a respeito da teoria de sistemas de controle, é indispensável um manual que descreva detalhadamente as funções e propriedades de qualquer software destinado à área de Engenharia de Controle, tornando seu uso fácil e correto. A seguir, o usuário terá acesso aos nomes dos acessórios, funcionamento e descrição de cada parte da ferramenta desenvolvida neste trabalho.

A.1 Objetos de uma interface gráfica

Segue abaixo os principais objetos utilizados para a construção das interfaces da ferramenta, devidamente identificados na figura (A.1).

1. **Menu Bar**: contém alguns itens cujas funções podem ser selecionadas.
2. **Push Button**: quando se deseja executar uma função já programada, basta clicar nesse botão.
3. **Edit Text**: permite o usuário entrar com dados. Neste trabalho, foi utilizado também como saída de alguns dados.
4. **Axes**: área da interface onde são plotados os gráficos das simulações.
5. **Radio Button**: marcadores que podem ou não ser selecionados. Esse aplicativo é útil quando o usuário deve selecionar um item dentro de algumas opções.

6. **Checkbox**: é bastante semelhante ao “radiobutton”. Permite a seleção de um ou mais itens desejados. Também é possível que nenhum “checkbox” seja selecionado.

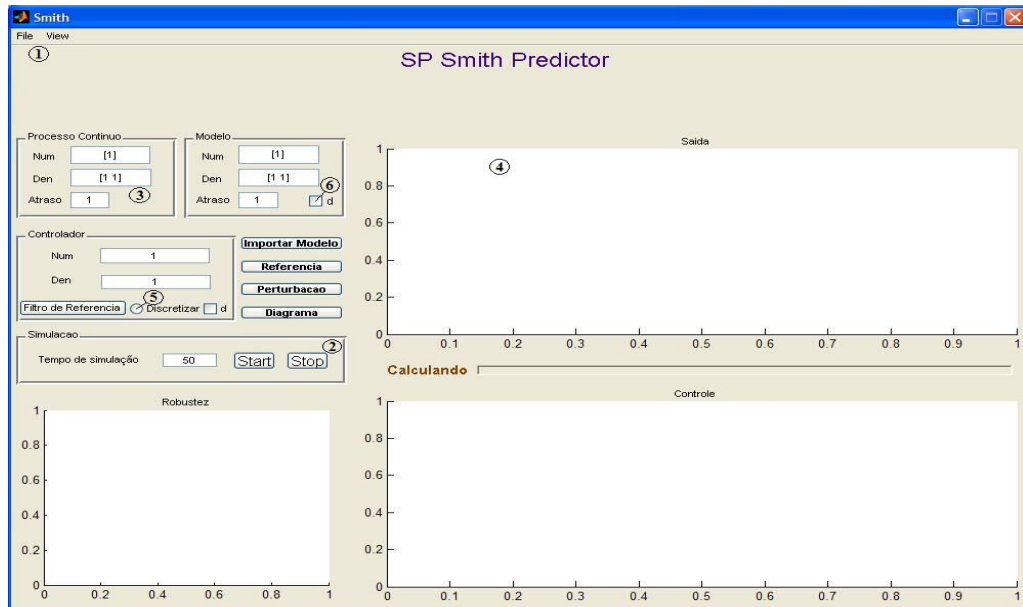


Figura A.1: Elementos de uma interface

A.2 Instalando e iniciando a ferramenta

Para que a ferramenta seja utilizada, é necessário que o usuário tenha o Matlab instalado no seu computador. Os arquivos da ferramenta devem ser salvos na pasta onde deseja-se trabalhar.

Para começar o uso da ferramenta é muito simples. É preciso apenas abrir o arquivo “home.m” através do Matlab e clicar em *Run*. Desta forma, abrirá a Interface Inicial onde o usuário terá acesso às partes de Identificação, Controladores e Resultados da ferramenta.

A.3 Ferramenta de identificação

Na parte de identificação, existem algumas opções que são comuns nos métodos utilizados: *Importar dados*, *Ruído*, *Validação* e *Exportar modelo*. É importante lembrar que o formato dos dados de entrada e saída é o mesmo do próprio Matlab. A interface padrão de qualquer parte da ferramenta já vem com a maioria dos “Edit Text’s” preenchida no sentido de orientar o usuário.

Clicando em *Importar dados*, abrirá uma janela onde a ferramenta pode carregar um arquivo que contém os dados de entrada e saída de um processo real e fazer a sua devida plotagem. O arquivo deve estar no formato “.mat” e com a 1ª coluna da matriz com os dados de entrada e a 2ª com os de saída. Desta forma, a identificação deve ser feita, definindo os parâmetros indicados pela ferramenta e clicando em *Identificação* ou *Modelo*. Este último para o método gráfico. *OBS*¹: independentemente do nome do arquivo (“nome.mat”), a matriz que contém os dados deve ser nomeada como “medidas”. *OBS*²: o *Sinal de entrada* não pode ser clicado. Caso seja, é necessário importar novamente os dados.

Clicando em *Ruído*, a ferramenta permite configurar um ruído que afeta a leitura de saída do processo simulado. Abrirá uma nova janela onde o usuário pode digitar a amplitude deste ruído e a função de transferência de sua filtragem.

Clicando em *Validação*, abrirá uma nova janela onde o usuário pode simular uma validação do último modelo obtido. O tempo padrão será três vezes maior que o tempo utilizado na identificação de um processo simulado, entretanto o usuário pode modificar. Nesta nova janela, basta o usuário clicar em *Simular* para fazer uma nova plotagem e calcular o seu índice de erro quadrático. Caso o usuário tenha importado os dados reais de um determinado processo, ao invés de tempo, o número de pontos que deverá ser ajustado. O número padrão corresponde ao total de pontos disponíveis do arquivo importado.

Clicando em *Exportar modelo*, a ferramenta armazena na memória o último modelo obtido para ser usado no projeto dos controladores.

A.3.1 Método Gráfico

Primeiramente, o usuário deve preencher os dados da função de transferência do processo. Caso o “Checkbox” *d* esteja selecionado, estes dados serão lidos como parâmetros de um processo discreto. Caso contrário, parâmetros de um processo contínuo. Feito isso, deve-se escolher o tempo de simulação e de amostragem e clicar em *Step* para aplicar um degrau unitário da entrada do processo. Assim, alguns dados serão fornecidos pela ferramenta juntamente com a plotagem do processo. Estes dados são os valores de saída do processo, os quais suas respectivas constantes de tempo deverão ser buscadas pelo usuário.

O usuário deve, então, preencher os “Edit Text’s” correspondentes ao modelo desejado:

- (i) Para o modelo contínuo de 1ª ordem, é necessário preencher o $Y_{inicial}$, o Y_{final} e suas

constantes de tempo, $T_{128.3\%}$ e $T_{263.2\%}$. Para o modelo contínuo de 2ª ordem, é necessário preencher o $Y_{inicial}$, o Y_{final} e suas constantes de tempo, $T_{15\%}$, $T_{45\%}$ e $T_{75\%}$.

Clicando em *Modelo*, os parâmetros do modelo contínuo serão fornecidos e uma nova plotagem é feita, comparando o modelo obtido com o processo real.

Uma particularidade ocorre quando o usuário opta por uma importação de dados. Neste método, uma nova janela será aberta. Nesta janela, deve-se preencher os dados, indicados pela ferramenta, do degrau aplicado na entrada e clicar em “Ok”. Feito isso, os “Edit Text’s” devem ser preenchidos como no caso de um processo simulado. É importante o usuário estar atento à qualidade dos dados importados para uma boa identificação.

A.3.2 Métodos dos Mínimos Quadrados e dos Mínimos Quadrados Recursivo

Os dois métodos funcionam de forma similar. As únicas diferenças são no ganho de ponderação λ que existe no MMQ recursivo e nos modelos fornecidos.

Igualmente ao método gráfico, os dados da função de transferência do modelo contínuo devem ser preenchidos. Feito isso, os dados do sinal do tipo PRBS que será aplicado na entrada do processo também devem ser preenchidos. Para aplicar o sinal, basta clicar em *Sinal de entrada* e, clicando em *Identificação*, os modelos são fornecidos para cada atraso estimado.

OBS: Para cada atraso estimado, não é necessário aplicar um novo sinal de entrada, exceto se o sinal de entrada não foi muito bem escolhido.

A.4 Ferramenta de Controle

Na ferramenta de controle existem alguns “Push Button’s” básicos que são comuns nos controladores:

Clicando em *Importar Modelo*, a ferramenta carrega o modelo disponível na sua memória, obtido pelo processo de identificação. Caso o “Checkbox” d do modelo esteja selecionado, o modelo discretizado é carregado.

Clicando em *Referência*, uma janela para a configuração da referência será aberta. Esta opção permite a escolha de até três mudanças nas amplitudes da referência. Para isto, basta preencher os valores das amplitudes com seus respectivos tempos de aplicação.

Clicando em *Perturbação*, uma janela para a configuração da perturbação será aberta. Nesta nova janela, o usuário pode escolher o tipo de perturbação a ser aplicado e abrir uma nova janela para a configuração da função de transferência da perturbação. Caso o usuário apenas escolha o tipo da perturbação, sua função de transferência assume seu valor padrão, $Q(s) = 1$.

Clicando em *Diagrama*, uma figura com diagramas de blocos que representa a implementação do sistema é aberta.

Clicando em *Start*, inicia a simulação.

Clicando em *Stop*, interrompe a simulação.

Após preencher os dados do processo e do modelo (este último pode ser importado!), o usuário deve ajustar o controlador. Um “Menu Bar” que permite salvar resultados e uma melhor visualização das plotagens, é o mesmo para todos os controladores.

A.4.1 Controlador PID

Existem três configurações implementadas na ferramenta para o PID: acadêmica, série e paralela. Para escolher, basta clicar no “Radio Button” desejado.

Clicando em *ZN*, *Cohen*, *S-IMC* ou *PS*, o usuário opta por um ajuste automático. No *PS*, o ajuste é feito em função do tempo de resposta desejado T_0 .

Na área dos *Filtros*, o usuário pode configurar um filtro de referência no sistema de controle, clicando em *Referência*; e ajustar o parâmetro α do filtro da parte derivativa do controlador mostrado nas equações (2.12), (2.13) e (2.14).

O usuário também pode implementar o controle no tempo discreto. Para isto, é só clicar no “Checkbox” *Discretizar PID* e uma janela onde deve ser escolhido o período de amostragem, será aberta. Em todos os controladores a discretização é feita pelo método de Tustin. Caso o usuário deseje fazer o ajuste diretamente no discreto, existe a área *PID Discreto* e um “Radio Button” que deve ser selecionado.

IMPORTANTE: embora a ferramenta tenha um sistema de ajuda através de janelas de “Erro” e “Atenção”, é fundamental o usuário ficar atento no tipo de implementação do modelo para análise da robustez.

A.4.2 Preditor de Smith

O ajuste do PS é feito determinando as funções de transferência do controlador primário e do filtro de referência. A implementação do controle pode ser feita de três formas: (i) no tempo contínuo; (ii) no tempo discreto a partir de uma discretização do controlador, selecionando o “Radio Button” *Discretizar*; (iii) diretamente no tempo discreto, selecionando o “Checkbox” *d*.

A.4.3 Observador de Perturbações

Nesta versão da ferramenta, o DO-DTC limita-se a analisar sistemas de 1ª ou 2ª ordem. Este último podendo ter um zero. O ajuste do controle é feito determinando o filtro de referência e o parâmetro T_v do controlador, cujas implementações podem ser de duas formas: (i) no tempo contínuo; (ii) no tempo discreto a partir de uma discretização, selecionando o “Radio Button” *Discretizar*.

A.4.4 Preditor de Smith Filtrado

O ajuste do PSF é feito determinando as funções de transferência do controlador primário, do filtro de referência e do parâmetro α do filtro de robustez. O usuário tem também duas formas de ajuste automático: (i) clicando em *FOPDT*, o ajuste é programado para modelos de 1ª ordem com atraso e (ii) clicando em *IPDT*, o ajuste é programado para modelos de plantas integradoras. Em ambos os casos, o usuário deve definir o tempo de resposta em malha fechada desejado em Tr .

A implementação do controle é feita apenas no tempo discreto através de uma discretização já programada ou um ajuste direto no tempo discreto, clicando no “Checkbox” *d*. O tempo de amostragem deve ser definido antes da simulação.

A.4.5 GPC

O GPC tem como parâmetros de ajuste o *Horizonte de predição*, *Horizonte de controle*, o valor de λ e o *Polinômio T*. O algoritmo de controle foi desenvolvido para implementar numa estrutura de DTC discreto, portanto só é válido para modelos que tenham um atraso $dn \geq 1$.

A.4.6 DTC-GPC

O DTC-GPC tem como parâmetros de ajuste o *Horizonte de predição*, *Horizonte de controle*, o valor de λ , *Polinômio T* e o valor de β do filtro de robustez. Como o controlador é implementado no tempo discreto, β deve ser menor que 1.

A.4.7 Salvando resultados

Em cada controlador, o usuário pode salvar até cinco plotagens para comparar resultados. A razão do limite ser cinco é de evitar uma poluição visual. Com mais de cinco resultados salvos, as plotagens vão se renovando em ordem, à medida que forem salvas. Por exemplo: se o usuário salvar sete resultados, a partir da 5ª plotagem, as primeiras (1ª e 2ª) darão lugar às últimas salvas (6ª e 7ª), automaticamente. Surgirão também legendas sugestivas e que podem ser alteradas em cada plotagem, conforme a preferência do usuário. Isto pode ser feito para cada simulação seguindo a ordem: *File* \rightarrow *Salvar* \rightarrow *Plotar resultados*.

Clicando em *Limpar*, apagará a memória que contém as plotagens.

Com o botão *Exportar resultado* ativo, a última plotagem para cada controlador ficará disponível para outro tipo de comparação que pode ser feita na seção *Resultados* da ferramenta.

A.5 Comparando resultados finais

Clicando em *Resultados* na Interface Inicial, o usuário terá acesso a uma seção bastante útil. Esta parte da ferramenta deve ser utilizada na comparação de resultados armazenados dos controladores. O usuário tem a total liberdade de definir quais os controladores lhe convém comparar. A ferramenta fornecerá a plotagem e os índices, ISE e IAE, somente de cada controlador selecionado, através do seu respectivo “Checkbox”.

Referências Bibliográficas

- [1] Lelic M. A. and Wellstead P. E. Generalized pole placement self tuning controller. part 1. basic algorithm. pages 547–568. Int. J. Control, 46(2), 1987.
- [2] Linkers D. A and Mahfont M. Advances in model-based predictive control, chapter generalized predictive control in clinical anaesthesia. Oxford University Press, 1994.
- [3] Karl Johan Aström. Pid controllers: Theory, design and tuning. 2nd edn, Instrument Society of America, 1995.
- [4] Karl Johan Aström. Process control - past, present and future. volume 5, pages 3–10. IEEE - Control Systems Magazine, ago. 1985.
- [5] Karl Johan Aström and T. Hägglund. Advanced pid control. ISA - The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC 27709, 2005.
- [6] E. F. Camacho and C. Bordons. Model predictive control. Springer-Verdag, 2004.
- [7] Clarke D. Application of generalized prective control to industrial processes. pages 49–55. IEEE Control Systems Magazine,122, 1988.
- [8] Clarke D. and Mothadi C. Properties of generalized predictive control. pages 859–875. Automatica, 25(6), 1989.
- [9] Clarke D., Mothadi C., and Tuffs P. Generalized predictive control. part i the basic algorithm and part ii extensions and interpretations. pages 137–160. Automatica, 22(2), 1987.
- [10] N. Duro, R. Dormido, H. Vargas, S. Dormido-Canto, J. Sánchez, G. Farias, and S. Dormido. An integrated virtual and remote control lab: The three-tank system as a case study. volume 10, pages 50–59. Computing in Science & Engineering, 2008.

- [11] Fontanini, O. S. S. Filho, and P. A. V. Ferreira. Um ambiente integrado para análise e projeto no espaço de estados. volume 1, pages 189–194, Belém/PA, 1990. Anais do 8°. CBA - Congresso Brasileiro de Automática.
- [12] Cohen G. and Coon G. Theoretical consideration of retarded control. pages 827–834. Trans. ASME, 1975.
- [13] Zigler J. G. and Nichols N. B. Optimum settings for automatic controllers. pages 759–768. Trans. ASME (64), 1942.
- [14] P. García, P. Albertos, and T. Hägglund. Control of unstable nonminimum-phase delayed systems. volume 16, pages 1099–1111. Journal of Process Control, 2006.
- [15] H. Garnier, M. Gilson, and W. Zheng. A bias-eliminated least-squares method for continuous-time model identification of closed-loop systems. pages 38–48. International Journal of Control 73(1), 2000.
- [16] S. M. Guo, W. Wang, and L. S. Shieh. Discretisation of two degree-of-freedom controller and system with state, and output delays. pages 87–96. IEE Proceedings Control Theory and Applications, 147(1), 2000.
- [17] W. K. Ho, T. H. Lee, and O. P. Gan. Tuning of multiloop pid controllers based on gain and phase margin specifications. pages 2231–2238. Ind. Eng. Chem. Res., 36, 1997.
- [18] A. Ingimundarson and T. Hägglund. Performance comparison between pid and dead-time compensating controllers. volume 12, pages 887–895. Journal of Process Control, 2002.
- [19] Carlos A. V. J. Desenvolvimento em interface gráfica em ambiente matlab. CIP-Brasil, 2005.
- [20] Palmor Z. J. Stability properties of smith dead time compensator controller. pages 937–949. Int. J. Control (32), 1980.
- [21] L. Ljung. System identification. theory for the user. Prentice-Hall, 1987.
- [22] X. Lu, Y. S. Yang, Q. G. Wang, and W. X. Zheng. A double two-degree-of-freedom control scheme for improved control of unstable delay processes. volume 15, pages 605–614. Journal of Process Control, 2005.

- [23] M. R. Matausek and A. D. Micic. A modified smith predictor for controlling a process with a integrator and long dead-time. volume 44, pages 1603–1606. IEEE Transactions on Automatic Control, 1999.
- [24] W. Michiels and S. I. Niculescu. On the delay sensivity of smith predictors. volume 34 (8-9), pages 543–552. International Journal of Systems Science, 2003.
- [25] J. E. Normey-Rico and E. F. Camacho. *Control of Dead-time Processes*. Springer, 2007.
- [26] J. E. Normey-Rico and E. F. Camacho. *Robustness effects of a prefilter in Smith predictor based generalised predictive controller*. IEE Proceedings, Control Theory and Applications 146(2):179-185, 1999.
- [27] J. E. Normey-Rico and E. F. Camacho. A unified approach to design dead-time compensators for stable and integrative process with dead-time. volume 47, pages 299–305. IEEE Transactions on Automatic Control, 2002.
- [28] J. E. Normey-Rico and E. F. Camacho. *Dead-time compensators: A survey*, volume 15 p.doi:10.1016. Control Engineering Practice, 2007.
- [29] J. E. Normey-Rico, C. Bordons, and E. F. Camacho. Improving the robustness of dead-time compensating pi controllers. volume 5 (6), pages 801–810. Control Engineering Practice, 1997.
- [30] Ansay P. and Wertz V. Model uncertainties in gpc: a systematic two-step design. Proc. of ECC 97, 1997.
- [31] Z. J. Palmor. Time delay compensation-smith predictor and modificationis, in the control handbook, (levine, w.s.,ed. pages 224–237. CRC Press, 1996.
- [32] Cutler C. R. and Ramaker B. C. Dynamic matrix control - a computer control algorithm. In Automatic Control Conference, 1980.
- [33] D.E. Rivera, M. Morari, and S. Skogestad. Internal model control 4. pid controller design industrial & engineering chemistry process design and development 25. pages 252–265, 1986.
- [34] Skogestad S. Simple analytic rules for model reduction and pid controller tuning. pages 291–309. J. of Process Control, 13, 2003.

- [35] B. R. Shearer, K. C. Daly, G. C. Goodwin, and P. Waite. A suite of programs for classical control systems design. pages 113–118, University of Cambridge, 1973. Proceedings of the IEE Conference on Computer Aided Control System Design.
- [36] G. J. Silva, A. Datta, and S. P. Bhattacharyya. Pid controllers for time delay systems. 2004.
- [37] S. Skogestad and I. Postlethwait. Multivariable feedback control. analysis and design. John Wiley and Sons, New York, 1996.
- [38] O. J. M. Smith. *Feedback Control Systems*. Mc Graw Hill, 1958.
- [39] Richalet T. Applications of model based predictive control. pages 1251–1274. Automatica, 29(5), 1993.
- [40] Richalet T., Rault A., Testud J. L., and Papon J. Algorithmic control of industrial processes. In 4th IFAC Symposium on Identification and System Parameter Estimation, 1976.
- [41] Richalet T., Rault A., Testud J. L., and Papon J. Model predictive heuristic control: Application to industrial processes. pages 413–428. Automatica, 14(2), 1978.
- [42] Robinson T. and Clarke D. Robustness effects of a prefilter in receding-horizon predictive control. pages 2–8. IEEE-D 138, 1991.
- [43] Yoon T. and Clarke D. Observer design in receding-horizon control. pages 171–191. Int. Journal of Control 2, 1995.
- [44] H. Takatsu, T. Itoh, and M. Araki. Future needs for control theory in industries-report and topics of the control technology survey in the japanese industry. pages 369–374. Journal of Process Control 8(5-6), 1998.
- [45] B. C. Torrico, R. De Keyser, C. Ionescu, and J. E. Normey-Rico. Robust predictive control of drug dosing during anesthesia. European Control Conference, 2007.
- [46] Q. G. Wang, H. Q. Zhou, and Y. Zhang. A comparative study on control of unstable processes with time delay. pages 2006–2014. 5th Asian Control Conference, Melbourne, Australia, 2004.
- [47] K. Watanabe and M. Ito. A process model control for linear systems with delay. volume 26 (6), pages 1261–1268. IEEE Transactions on Automatic Control, 1981.